

Zappy

1.0

Generated by Doxygen 1.9.1

1 Zappy	1
1.1 Table des matières	1
1.2 Vue d'ensemble	1
1.3 Architecture	2
1.4 Prérequis	2
1.5 Installation	2
1.6 Utilisation	2
1.6.1 Démarrer le serveur	2
1.6.2 Démarrer l'interface graphique	3
1.6.3 Démarrer l'IA	3
1.7 Composants	3
1.7.1 Serveur	3
1.7.2 Interface Graphique	4
1.7.3 Intelligence Artificielle	4
1.8 Développement	5
1.8.1 Structure du projet	5
1.8.2 Tests	5
2 Utilisation de DLLoader et des bibliothèques dynamiques graphiques dans Zappy GUI	7
2.1 Fonctionnement général	7
2.2 Exemple d'utilisation	7
2.3 Ajouter une nouvelle bibliothèque	7
2.4 Avantages	8
2.5 Exemple minimal	8
2.5.1 Compilation	8
2.5.2 Exécution	8
2.6 Fonctionnement	8
2.7 Pour aller plus loin	8
3 Serveur Zappy	9
3.1 Vue d'ensemble	9
3.2 Architecture	9
3.2.1 Composants Principaux	9
3.2.2 Structures de Données Clés	9
3.3 Compilation	9
3.4 Utilisation	9
3.4.1 Démarrage Standard	9
3.4.2 Paramètres	9
3.4.3 Exemples Avancés	10
3.5 Protocoles de Communication	10
3.5.1 Clients IA	10
3.5.2 Clients GUI	10
3.6 Performance et Optimisation	10

3.6.1 Métriques	10
3.6.2 Optimisations Implémentées	10
3.7 Tests	11
3.7.1 Tests Unitaires	11
3.7.2 Tests de Performance	11
3.8 Debugging	11
3.8.1 Mode Debug	11
3.8.2 Logs Verbeux	11
3.9 Architecture Interne	11
3.9.1 Boucle Principale	11
3.9.2 Gestion des Commandes	11
3.9.3 États des Joueurs	12
3.10 Configuration Avancée	12
3.10.1 Variables d'Environnement	12
3.10.2 Fichier de Configuration	12
3.11 Monitoring et Maintenance	12
3.11.1 Signaux Système	12
3.11.2 Métriques Runtime	12
3.12 Contribuer	12
3.12.1 Standards de Code	12
3.12.2 Workflow	13
3.13 Documentation Complète	13
4 RayGUICPP - Encapsulation C++ de raygui pour Zappy	15
4.1 Prérequis	15
4.2 Structure	15
4.3 Compilation	15
4.4 Exemple minimal	15
4.5 Utilisation dans Zappy	15
4.6 Intégration dynamique (DLLoader)	16
4.7 Exemple d'utilisation avec DLLoader	16
4.8 Astuces	16
4.9 Conseils d'extension	16
4.10 Liens utiles	16
5 RaylibCPP - Encapsulation C++ de Raylib	17
5.1 Fonctionnalités principales	17
5.2 Structure du dossier	17
5.3 Exemple d'utilisation	17
5.4 Intégration dynamique (DLLoader)	18
5.5 Exemple d'utilisation avec DLLoader	18
5.6 Notes avancées	18

6 Protocole de Communication Serveur-IA	19
6.1 Vue d'ensemble	19
6.2 Connexion	19
6.3 Format des commandes	19
6.4 Commandes disponibles	19
6.4.1 Commandes de base	19
6.4.2 Format des réponses	20
6.4.2.1 Look	20
6.4.2.2 Inventory	20
6.4.2.3 Broadcast	20
6.4.2.4 Incantation	20
6.5 Conditions d'élévation	20
6.6 Mort du joueur	20
6.7 Exemple d'échange	20
7 Documentation du Projet Zappy	23
7.1 Introduction	23
7.2 Vue d'ensemble du Système	23
7.2.1 Composants Principaux	23
7.2.2 Architecture Technique	23
7.3 Structure de la Documentation	23
7.3.1 Guides d'Utilisation	23
7.3.2 Documentation Technique	23
7.3.2.1 Composants Principaux	23
7.3.2.2 Bibliothèques et Outils	24
7.3.3 Protocoles de Communication	24
7.3.4 Documentation Avancée	24
7.4 Documentation Interactive	24
7.4.1 Docusaurus (Documentation Web)	24
7.4.2 Doxygen (Documentation API)	24
7.5 Démarrage Rapide	25
7.5.1 Installation des Dépendances	25
7.5.2 Compilation et Test	25
7.6 Contribution à la Documentation	25
7.6.1 Standards de Documentation	25
7.6.2 Workflow de Contribution	25
7.6.3 Organisation des Fichiers	25
7.7 Ressources Complémentaires	25
7.7.1 Références Externes	25
7.7.2 Support et Communauté	26
7.7.3 Licence et Copyright	26
8 Namespace Index	27

8.1 Namespace List	27
9 Hierarchical Index	29
9.1 Class Hierarchy	29
10 Data Structure Index	33
10.1 Data Structures	33
11 File Index	37
11.1 File List	37
12 Namespace Documentation	41
12.1 actions Namespace Reference	41
12.1.1 Function Documentation	41
12.1.1.1 go_take_item()	41
12.1.1.2 go_to_pos_with_distance()	42
12.1.1.3 got_to_dir()	42
12.1.1.4 take_all_of_item_here()	42
12.1.1.5 take_everything_here()	42
12.2 agent Namespace Reference	43
12.3 behaviors Namespace Reference	43
12.4 broadcastManager Namespace Reference	43
12.5 decisionManager Namespace Reference	43
12.6 encryption Namespace Reference	43
12.6.1 Function Documentation	43
12.6.1.1 decrypt_message()	44
12.6.1.2 encrypt_message()	44
12.6.2 Variable Documentation	44
12.6.2.1 secret_key	44
12.7 input Namespace Reference	44
12.7.1 Function Documentation	44
12.7.1.1 get_key()	44
12.8 logger Namespace Reference	45
12.9 main Namespace Reference	45
12.9.1 Function Documentation	45
12.9.1.1 run_agent()	45
12.9.1.2 signal_handler()	45
12.9.2 Variable Documentation	45
12.9.2.1 args	46
12.9.2.2 child_pids	46
12.9.2.3 ip	46
12.9.2.4 num_agents	46
12.9.2.5 pid	46
12.9.2.6 port	46

12.9.2.7 team	46
12.10 paringArgsClass Namespace Reference	46
12.11 RayGUICPP Namespace Reference	46
12.12 raylibcpp Namespace Reference	47
12.12.1 Typedef Documentation	48
12.12.1.1 RenderStatePtr	48
12.12.1.2 WaveTextConfigPtr	48
12.12.1.3 WaveTextConfigUPtr	48
12.13 raylibcpp::Text3DHelper Namespace Reference	48
12.13.1 Function Documentation	48
12.13.1.1 calculateAdvanceX()	49
12.13.1.2 calculateGlyphDimensions()	49
12.13.1.3 calculateGlyphPosition()	49
12.13.1.4 calculateScale()	49
12.13.1.5 calculateSourceRectangle()	49
12.13.1.6 calculateTextureCoordinates()	50
12.13.1.7 getGlyphIndex()	50
12.13.1.8 isNewline()	50
12.13.1.9 isRenderableCharacter()	50
12.13.1.10 isValidFont()	50
12.13.1.11 isValidRenderParams()	50
12.13.1.12 isWhitespace()	51
12.13.2 Variable Documentation	51
12.13.2.1 MAX_FONT_SIZE	51
12.13.2.2 MAX_SPACING	51
12.13.2.3 MIN_FONT_SIZE	51
12.13.2.4 MIN_SPACING	51
12.14 resources Namespace Reference	51
12.14.1 Variable Documentation	51
12.14.1.1 resources	51
12.15 roles Namespace Reference	52
12.15.1 Variable Documentation	52
12.15.1.1 roles	52
12.16 socketManager Namespace Reference	52
12.17 SystemWrapper Namespace Reference	52
12.17.1 Function Documentation	53
12.17.1.1 closeSocket()	53
12.17.1.2 connectSocket()	53
12.17.1.3 createSocket()	53
12.17.1.4 executeWithTimeout()	53
12.17.1.5 getErrorString()	54
12.17.1.6 getSocketOption()	54

12.17.1.7 inetNtop()	54
12.17.1.8 inetPton()	54
12.17.1.9 makeSharedArray()	54
12.17.1.10 makeUniqueArray()	55
12.17.1.11 pollSocket() [1/2]	55
12.17.1.12 pollSocket() [2/2]	55
12.17.1.13 readSocket()	55
12.17.1.14 setNonBlocking()	55
12.17.1.15 writeSocket()	56
12.18 TypeAdapter Namespace Reference	56
12.18.1 Function Documentation	56
12.18.1.1 FromRaylib() [1/4]	56
12.18.1.2 FromRaylib() [2/4]	56
12.18.1.3 FromRaylib() [3/4]	57
12.18.1.4 FromRaylib() [4/4]	57
12.18.1.5 ToRaylib() [1/4]	57
12.18.1.6 ToRaylib() [2/4]	57
12.18.1.7 ToRaylib() [3/4]	57
12.18.1.8 ToRaylib() [4/4]	58
12.19 upgrades Namespace Reference	58
12.20 zappy Namespace Reference	58
12.20.1 Function Documentation	58
12.20.1.1 get_best_available_resource()	58
12.20.1.2 get_closest_of_item()	58
12.20.1.3 get_item_relative_pos()	59
12.20.1.4 go_get_item()	59
12.20.1.5 how_much_of_item_here()	59
12.20.1.6 parse_inventory()	59
12.21 Zappy Namespace Reference	60
12.22 ZappyTypes Namespace Reference	60
12.22.1 Enumeration Type Documentation	61
12.22.1.1 KeyboardKey	61
12.23 ZappyTypes::Colors Namespace Reference	62
12.23.1 Variable Documentation	63
12.23.1.1 Z_BEIGE	63
12.23.1.2 Z_BLACK	63
12.23.1.3 Z_BLANK	63
12.23.1.4 Z_BLUE	63
12.23.1.5 Z_BROWN	63
12.23.1.6 Z_DARKBLUE	64
12.23.1.7 Z_DARKBROWN	64
12.23.1.8 Z_DARKGRAY	64

12.23.1.9 Z_DARKGREEN	64
12.23.1.10 Z_DARKPURPLE	64
12.23.1.11 Z_GOLD	64
12.23.1.12 Z_GRAY	64
12.23.1.13 Z_GREEN	64
12.23.1.14 Z_LIGHTGRAY	64
12.23.1.15 Z_LIME	65
12.23.1.16 Z_MAGENTA	65
12.23.1.17 Z_MAROON	65
12.23.1.18 Z_ORANGE	65
12.23.1.19 Z_PINK	65
12.23.1.20 Z_PURPLE	65
12.23.1.21 Z_RAYWHITE	65
12.23.1.22 Z_RED	65
12.23.1.23 Z_SKYBLUE	66
12.23.1.24 Z_VIOLET	66
12.23.1.25 Z_WHITE	66
12.23.1.26 Z_YELLOW	66
13 Data Structure Documentation	67
13.1 AException Class Reference	67
13.1.1 Detailed Description	68
13.1.2 Constructor & Destructor Documentation	68
13.1.2.1 AException()	68
13.1.2.2 ~AException()	68
13.1.3 Member Function Documentation	68
13.1.3.1 getFormattedMessage()	69
13.1.3.2 getMessage()	69
13.1.3.3 getType()	69
13.1.3.4 what()	69
13.2 agent.Agent Class Reference	69
13.2.1 Detailed Description	70
13.2.2 Constructor & Destructor Documentation	70
13.2.2.1 __init__()	70
13.2.3 Member Function Documentation	70
13.2.3.1 get_message()	71
13.2.3.2 has_messages()	71
13.2.3.3 process_server_message()	71
13.2.3.4 run()	71
13.2.3.5 send_command()	71
13.2.3.6 start()	72
13.2.3.7 stop()	72

13.2.4 Field Documentation	72
13.2.4.1 broadcastManager	72
13.2.4.2 currentBehaviour	72
13.2.4.3 decisionManager	73
13.2.4.4 id	73
13.2.4.5 ip	73
13.2.4.6 level	73
13.2.4.7 logger	73
13.2.4.8 map_size_x	73
13.2.4.9 map_size_y	73
13.2.4.10 port	73
13.2.4.11 sock	73
13.2.4.12 socketManager	73
13.2.4.13 team	74
13.3 raylibcpp::Audio Class Reference	74
13.3.1 Detailed Description	74
13.3.2 Constructor & Destructor Documentation	74
13.3.2.1 Audio()	74
13.3.2.2 ~Audio()	74
13.4 behaviors.Behavior Class Reference	75
13.4.1 Detailed Description	75
13.4.2 Constructor & Destructor Documentation	75
13.4.2.1 __init__()	76
13.4.3 Member Function Documentation	76
13.4.3.1 execute()	76
13.4.4 Field Documentation	76
13.4.4.1 agent	76
13.5 BroadcastData Class Reference	76
13.5.1 Detailed Description	77
13.5.2 Constructor & Destructor Documentation	77
13.5.2.1 BroadcastData()	77
13.5.3 Member Function Documentation	77
13.5.3.1 getMessage()	77
13.5.3.2 getPlayerId()	77
13.5.3.3 getType()	78
13.5.3.4 setMessage()	78
13.5.3.5 setPlayerId()	78
13.6 broadcastManager.BroadcastManager Class Reference	78
13.6.1 Detailed Description	78
13.6.2 Constructor & Destructor Documentation	78
13.6.2.1 __init__()	78
13.6.3 Member Function Documentation	78

13.6.3.1 manage_broadcast()	79
13.6.4 Field Documentation	79
13.6.4.1 agent	79
13.7 RayGUICPP::Button Class Reference	79
13.7.1 Detailed Description	79
13.7.2 Member Function Documentation	80
13.7.2.1 Draw()	80
13.8 raylibcpp::Camera2DWrap Class Reference	80
13.8.1 Detailed Description	80
13.8.2 Constructor & Destructor Documentation	80
13.8.2.1 Camera2DWrap()	80
13.8.2.2 ~Camera2DWrap()	80
13.8.3 Member Function Documentation	80
13.8.3.1 get()	81
13.9 raylibcpp::Camera3DWrap Class Reference	81
13.9.1 Detailed Description	81
13.9.2 Constructor & Destructor Documentation	81
13.9.2.1 Camera3DWrap() [1/2]	81
13.9.2.2 Camera3DWrap() [2/2]	81
13.9.2.3 ~Camera3DWrap()	82
13.9.3 Member Function Documentation	82
13.9.3.1 beginMode()	82
13.9.3.2 endMode()	82
13.9.3.3 get()	82
13.10 CameraController Class Reference	82
13.10.1 Detailed Description	83
13.10.2 Constructor & Destructor Documentation	83
13.10.2.1 CameraController()	83
13.10.2.2 ~CameraController()	83
13.10.3 Member Function Documentation	83
13.10.3.1 angleX()	83
13.10.3.2 angleY()	83
13.10.3.3 distance()	83
13.10.3.4 handleKeyboardInput()	84
13.10.3.5 handleMouseInput()	84
13.10.3.6 init()	85
13.10.3.7 isDragging()	85
13.10.3.8 position()	85
13.10.3.9 reset()	85
13.10.3.10 setDragging()	85
13.10.3.11 setMapDimensions()	85
13.10.3.12 setMousePosition()	86

13.10.3.13 target()	86
13.10.3.14 up()	86
13.10.3.15 update()	86
13.10.3.16 updateCameraPosition()	86
13.11 RayGUICPP::CheckBox Class Reference	87
13.11.1 Detailed Description	87
13.11.2 Member Function Documentation	87
13.11.2.1 Draw()	87
13.12 circular_buffer_s Struct Reference	87
13.12.1 Detailed Description	87
13.12.2 Field Documentation	87
13.12.2.1 buffer	87
13.12.2.2 count	88
13.12.2.3 end	88
13.12.2.4 start	88
13.13 client_s Struct Reference	88
13.13.1 Detailed Description	88
13.13.2 Field Documentation	89
13.13.2.1 client_add	89
13.13.2.2 client_fd	89
13.13.2.3 client_id	89
13.13.2.4 client_poll	89
13.13.2.5 is_fully_connected	89
13.13.2.6 next	89
13.13.2.7 player	89
13.13.2.8 type	89
13.14 ZappyTypes::Color Struct Reference	89
13.14.1 Detailed Description	90
13.14.2 Field Documentation	90
13.14.2.1 a	90
13.14.2.2 b	90
13.14.2.3 g	90
13.14.2.4 r	90
13.15 RayGUICPP::ColorPicker Class Reference	90
13.15.1 Detailed Description	90
13.15.2 Member Function Documentation	90
13.15.2.1 Draw()	91
13.16 RayGUICPP::ComboBox Class Reference	91
13.16.1 Detailed Description	91
13.16.2 Member Function Documentation	91
13.16.2.1 Draw()	91
13.17 command_data_s Struct Reference	91

13.17.1 Detailed Description	91
13.17.2 Field Documentation	92
13.17.2.1 accepted_types	92
13.17.2.2 commands	92
13.17.2.3 functions	92
13.17.2.4 times	92
13.18 decisionManager.DecisionManager Class Reference	92
13.18.1 Detailed Description	92
13.18.2 Constructor & Destructor Documentation	92
13.18.2.1 __init__()	92
13.18.3 Member Function Documentation	93
13.18.3.1 take_action()	93
13.18.4 Field Documentation	93
13.18.4.1 agent	93
13.18.4.2 behaviors	93
13.19 Zappy::DetailedTileRenderStrategy Class Reference	93
13.19.1 Detailed Description	94
13.19.2 Constructor & Destructor Documentation	94
13.19.2.1 DetailedTileRenderStrategy()	94
13.19.3 Member Function Documentation	95
13.19.3.1 renderTile()	95
13.20 DLLoader< T > Class Template Reference	95
13.20.1 Detailed Description	96
13.20.2 Constructor & Destructor Documentation	96
13.20.2.1 DLLoader()	96
13.20.2.2 ~DLLoader()	96
13.20.3 Member Function Documentation	96
13.20.3.1 getInstance()	96
13.20.3.2 getSymbol()	97
13.21 RayGUICPP::DropdownBox Class Reference	97
13.21.1 Detailed Description	97
13.21.2 Member Function Documentation	98
13.21.2.1 Draw()	98
13.22 behaviors.DysonBehavior Class Reference	98
13.22.1 Detailed Description	99
13.22.2 Member Function Documentation	99
13.22.2.1 execute()	99
13.23 egg_s Struct Reference	99
13.23.1 Detailed Description	100
13.23.2 Field Documentation	100
13.23.2.1 egg_id	100
13.23.2.2 next	100

13.23.2.3 pos_x	100
13.23.2.4 pos_y	100
13.23.2.5 team_name	100
13.24 EggData Class Reference	101
13.24.1 Detailed Description	102
13.24.2 Member Enumeration Documentation	102
13.24.2.1 EggAction	102
13.24.3 Constructor & Destructor Documentation	102
13.24.3.1 EggData() [1/2]	102
13.24.3.2 EggData() [2/2]	102
13.24.4 Member Function Documentation	102
13.24.4.1 getAction()	102
13.24.4.2 getEggId()	103
13.24.4.3 getPlayerId()	103
13.24.4.4 getType()	103
13.24.4.5 getX()	103
13.24.4.6 getY()	103
13.24.4.7 setEggId()	103
13.24.4.8 setPlayerId()	103
13.24.4.9 setX()	103
13.24.4.10 setY()	104
13.25 EndGameData Class Reference	104
13.25.1 Detailed Description	104
13.25.2 Constructor & Destructor Documentation	105
13.25.2.1 EndGameData()	105
13.25.3 Member Function Documentation	105
13.25.3.1 getTeamName()	105
13.25.3.2 getType()	105
13.25.3.3 setTeamName()	105
13.26 raylibcpp::Font Class Reference	105
13.26.1 Detailed Description	105
13.26.2 Constructor & Destructor Documentation	105
13.26.2.1 Font()	106
13.26.2.2 ~Font()	106
13.26.3 Member Function Documentation	106
13.26.3.1 drawText()	106
13.26.3.2 get()	106
13.27 GameLoop Class Reference	106
13.27.1 Detailed Description	107
13.27.2 Constructor & Destructor Documentation	107
13.27.2.1 GameLoop()	107
13.27.2.2 ~GameLoop()	108

13.27.3 Member Function Documentation	108
13.27.3.1 init()	108
13.27.3.2 onMapSizeChanged()	108
13.27.3.3 onTileChanged()	108
13.27.3.4 renderCube()	108
13.27.3.5 run()	109
13.27.3.6 setServerInfo()	109
13.28 behaviors.GetFoodAndMineralsBehavior Class Reference	109
13.28.1 Detailed Description	110
13.28.2 Member Function Documentation	110
13.28.2.1 execute()	110
13.29 behaviors.GetFoodBehavior Class Reference	111
13.29.1 Detailed Description	111
13.29.2 Member Function Documentation	111
13.29.2.1 execute()	112
13.30 behaviors.GetMineralsBehavior Class Reference	112
13.30.1 Detailed Description	113
13.30.2 Member Function Documentation	113
13.30.2.1 execute()	113
13.31 raylibcpp::GlyphDimensions Struct Reference	113
13.31.1 Detailed Description	113
13.31.2 Field Documentation	113
13.31.2.1 height	113
13.31.2.2 width	113
13.32 graphical_client_s Struct Reference	114
13.32.1 Detailed Description	114
13.32.2 Field Documentation	114
13.32.2.1 client	114
13.32.2.2 next	114
13.33 GraphicalContext Class Reference	114
13.33.1 Detailed Description	115
13.33.2 Member Typedef Documentation	115
13.33.2.1 UpdateFunction	115
13.33.3 Constructor & Destructor Documentation	115
13.33.3.1 GraphicalContext()	115
13.33.3.2 ~GraphicalContext()	116
13.33.4 Member Function Documentation	116
13.33.4.1 addObserver()	116
13.33.4.2 getDominantResourceType()	116
13.33.4.3 getMapHeight()	117
13.33.4.4 getMapWidth()	117
13.33.4.5 getPlayerInfo()	117

13.33.4.6 getResourceQuantity()	117
13.33.4.7 getTileData()	118
13.33.4.8 isPlayerOnTile()	118
13.33.4.9 removeObserver()	119
13.33.4.10 updateContext()	119
13.34 RayGUICPP::GroupBox Class Reference	119
13.34.1 Detailed Description	119
13.34.2 Member Function Documentation	119
13.34.2.1 Draw()	120
13.35 RayGUICPP::Icon Class Reference	120
13.35.1 Detailed Description	120
13.35.2 Member Function Documentation	120
13.35.2.1 IconText()	120
13.36 IException Class Reference	120
13.36.1 Detailed Description	121
13.36.2 Constructor & Destructor Documentation	121
13.36.2.1 ~IException()	121
13.36.3 Member Function Documentation	122
13.36.3.1 getFormattedMessage()	122
13.36.3.2 getMessage()	122
13.36.3.3 getType()	122
13.36.3.4 what()	122
13.37 IGraphicalContextObserver Class Reference	122
13.37.1 Detailed Description	123
13.37.2 Constructor & Destructor Documentation	123
13.37.2.1 ~IGraphicalContextObserver()	123
13.37.3 Member Function Documentation	123
13.37.3.1 onMapSizeChanged()	123
13.37.3.2 onTileChanged()	123
13.38 IGraphicsLib Class Reference	123
13.38.1 Detailed Description	125
13.38.2 Constructor & Destructor Documentation	125
13.38.2.1 ~IGraphicsLib()	125
13.38.3 Member Function Documentation	125
13.38.3.1 BeginCamera3D()	125
13.38.3.2 BeginDrawing()	125
13.38.3.3 BindTexture3D()	125
13.38.3.4 ClearBackground()	125
13.38.3.5 CloseWindow()	125
13.38.3.6 CreateCamera2D()	126
13.38.3.7 CreateCamera3D()	126
13.38.3.8 DrawCircle()	126

13.38.3.9 DrawCube()	126
13.38.3.10 DrawCylinder()	126
13.38.3.11 DrawGrid()	126
13.38.3.12 DrawLine3D()	127
13.38.3.13 DrawModel3D()	127
13.38.3.14 DrawModelEx()	127
13.38.3.15 DrawPlane()	127
13.38.3.16 DrawRectangle()	127
13.38.3.17 DrawSphere()	127
13.38.3.18 DrawText()	128
13.38.3.19 DrawText3D()	128
13.38.3.20 DrawTexture2D()	128
13.38.3.21 DrawTextWave3D()	128
13.38.3.22 EndCamera3D()	128
13.38.3.23 EndDrawing()	128
13.38.3.24 GetMousePosition()	129
13.38.3.25 GetMouseWheelMove()	129
13.38.3.26 GetMouseX()	129
13.38.3.27 GetMouseY()	129
13.38.3.28 InitWindow()	129
13.38.3.29 IsKeyDown()	129
13.38.3.30 IsKeyPressed()	129
13.38.3.31 IsMouseButtonDown()	129
13.38.3.32 IsMouseButtonPressed()	129
13.38.3.33 IsMouseButtonReleased()	130
13.38.3.34 IsTextureReady()	130
13.38.3.35 LoadFont()	130
13.38.3.36 LoadModel3D()	130
13.38.3.37 LoadModelWithTexture()	130
13.38.3.38 LoadTexture2D()	130
13.38.3.39 LoadTexture3D()	130
13.38.3.40 MeasureText3D()	130
13.38.3.41 PlayMusic()	131
13.38.3.42 PlaySound()	131
13.38.3.43 setFps()	131
13.38.3.44 SetMusicVolume()	131
13.38.3.45 SetSoundVolume()	131
13.38.3.46 StopMusic()	131
13.38.3.47 StopSound()	131
13.38.3.48 UnloadFont()	131
13.38.3.49 UnloadModel3D()	131
13.38.3.50 UnloadTexture2D()	132

13.38.3.51 UnloadTexture3D()	132
13.38.3.52 UpdateMusic()	132
13.38.3.53 WindowShouldClose()	132
13.39 IGuiLib Class Reference	132
13.39.1 Detailed Description	133
13.39.2 Constructor & Destructor Documentation	133
13.39.2.1 ~IGuiLib()	133
13.39.3 Member Function Documentation	133
13.39.3.1 ButtonPressed()	133
13.39.3.2 DrawButton()	134
13.39.3.3 DrawCheckBox()	134
13.39.3.4 DrawColorPicker()	134
13.39.3.5 DrawComboBox()	134
13.39.3.6 DrawDropdownBox()	134
13.39.3.7 DrawGroupBox()	135
13.39.3.8 DrawLabel()	135
13.39.3.9 DrawListView()	135
13.39.3.10 DrawMessageBox()	135
13.39.3.11 DrawPanel()	135
13.39.3.12 DrawProgressBar()	135
13.39.3.13 DrawScrollPanel()	136
13.39.3.14 DrawSlider()	136
13.39.3.15 DrawSpinner()	136
13.39.3.16 DrawStatusBar()	136
13.39.3.17 DrawTabBar()	137
13.39.3.18 DrawTextBox()	137
13.39.3.19 DrawTextInputBox()	137
13.39.3.20 DrawToggle()	137
13.39.3.21 DrawToggleGroup()	138
13.39.3.22 DrawValueBox()	138
13.39.3.23 DrawWindowBox()	138
13.39.3.24 GetStyle()	138
13.39.3.25 IconText()	138
13.39.3.26 LoadStyle()	138
13.39.3.27 SetStyle()	139
13.40 IMessageData Class Reference	139
13.40.1 Detailed Description	140
13.40.2 Constructor & Destructor Documentation	141
13.40.2.1 ~IMessageData()	141
13.40.3 Member Function Documentation	141
13.40.3.1 getType()	141
13.41 IncantationData Class Reference	141

13.41.1 Detailed Description	142
13.41.2 Constructor & Destructor Documentation	142
13.41.2.1 IncantationData()	142
13.41.3 Member Function Documentation	142
13.41.3.1 addPlayerId()	142
13.41.3.2 getLevel()	142
13.41.3.3 getPlayerIds()	142
13.41.3.4 getType()	143
13.41.3.5 getX()	143
13.41.3.6 getY()	143
13.41.3.7 setLevel()	143
13.41.3.8 setPlayerIds()	143
13.41.3.9 setX()	143
13.41.3.10 setY()	143
13.42 IncantationEndData Class Reference	143
13.42.1 Detailed Description	144
13.42.2 Constructor & Destructor Documentation	144
13.42.2.1 IncantationEndData()	144
13.42.3 Member Function Documentation	145
13.42.3.1 getType()	145
13.42.3.2 getX()	145
13.42.3.3 getY()	145
13.42.3.4 isSuccess()	145
13.42.3.5 setSuccess()	145
13.42.3.6 setX()	145
13.42.3.7 setY()	145
13.43 raylibcpp::Input Class Reference	146
13.43.1 Detailed Description	146
13.43.2 Member Function Documentation	146
13.43.2.1 getMousePosition()	146
13.43.2.2 getMouseWheelMove()	146
13.43.2.3 getX()	146
13.43.2.4 getY()	146
13.43.2.5 isKeyDown()	147
13.43.2.6 isKeyPressed()	147
13.43.2.7 isMouseButtonDown()	147
13.43.2.8 isMouseButtonPressed()	147
13.43.2.9 isMouseButtonReleased()	147
13.44 Zappy::ITileRenderStrategy Class Reference	147
13.44.1 Detailed Description	148
13.44.2 Constructor & Destructor Documentation	148
13.44.2.1 ~ITileRenderStrategy()	148

13.44.3 Member Function Documentation	148
13.44.3.1 renderTile()	148
13.45 RayGUICPP::Label Class Reference	148
13.45.1 Detailed Description	149
13.45.2 Member Function Documentation	149
13.45.2.1 Draw()	149
13.46 LibraryManager Class Reference	149
13.46.1 Detailed Description	149
13.46.2 Member Function Documentation	149
13.46.2.1 getGraphicsLib()	150
13.46.2.2 getGraphicsLibPtr()	150
13.46.2.3 getGuiLib()	150
13.46.2.4 getGuiLibPtr()	150
13.46.2.5 getInstance()	151
13.46.2.6 getLastErr()	151
13.46.2.7 loadGraphicsLib()	151
13.46.2.8 loadGuiLib()	151
13.47 RayGUICPP::ListView Class Reference	152
13.47.1 Detailed Description	152
13.47.2 Member Function Documentation	152
13.47.2.1 Draw()	152
13.48 logger.Logger Class Reference	153
13.48.1 Detailed Description	153
13.48.2 Constructor & Destructor Documentation	153
13.48.2.1 __init__()	153
13.48.3 Member Function Documentation	153
13.48.3.1 critical()	153
13.48.3.2 debug()	153
13.48.3.3 error()	154
13.48.3.4 info()	154
13.48.3.5 log()	154
13.48.3.6 warning()	154
13.48.4 Field Documentation	154
13.48.4.1 file	154
13.48.4.2 file_handle	154
13.48.4.3 message_prefix	155
13.49 logger.LogLevel Class Reference	155
13.49.1 Detailed Description	155
13.49.2 Field Documentation	155
13.49.2.1 CRITICAL	156
13.49.2.2 DEBUG	156
13.49.2.3 ERROR	156

13.49.2.4 INFO	156
13.49.2.5 WARNING	156
13.50 Zappy::MapRenderer Class Reference	156
13.50.1 Detailed Description	157
13.50.2 Constructor & Destructor Documentation	157
13.50.2.1 MapRenderer()	157
13.50.3 Member Function Documentation	157
13.50.3.1 initialize()	157
13.50.3.2 render()	157
13.50.3.3 setDetailThreshold()	158
13.50.3.4 setResourceColor()	158
13.50.3.5 setResourceRenderStrategy()	158
13.50.3.6 setTileRenderStrategy()	159
13.50.3.7 setTileSize()	159
13.50.3.8 setTileSpacing()	159
13.50.3.9 setZoomLevel()	159
13.51 Zappy::MapRendererObserver Class Reference	160
13.51.1 Detailed Description	160
13.51.2 Constructor & Destructor Documentation	160
13.51.2.1 MapRendererObserver()	161
13.51.3 Member Function Documentation	161
13.51.3.1 onMapSizeChanged()	161
13.51.3.2 onTileChanged()	161
13.52 MapSizeData Class Reference	161
13.52.1 Detailed Description	162
13.52.2 Constructor & Destructor Documentation	162
13.52.2.1 MapSizeData()	162
13.52.3 Member Function Documentation	162
13.52.3.1 getHeight()	162
13.52.3.2 getType()	162
13.52.3.3 getWidth()	163
13.52.3.4 setHeight()	163
13.52.3.5 setWidth()	163
13.53 Message Class Reference	163
13.53.1 Detailed Description	164
13.53.2 Constructor & Destructor Documentation	164
13.53.2.1 Message() [1/2]	164
13.53.2.2 Message() [2/2]	164
13.53.2.3 ~Message()	164
13.53.3 Member Function Documentation	164
13.53.3.1 addIntParam()	164
13.53.3.2 addStringParam()	164

13.53.3.3 getData()	165
13.53.3.4 getHeader()	165
13.53.3.5 getMessage()	165
13.53.3.6 getStructuredData()	165
13.53.3.7 hasStructuredData()	165
13.53.3.8 setData() [1/2]	165
13.53.3.9 setData() [2/2]	166
13.53.3.10 setHeader()	166
13.53.3.11 setMessage()	166
13.54 RayGUICPP::MessageBox Class Reference	166
13.54.1 Detailed Description	166
13.54.2 Member Function Documentation	166
13.54.2.1 Draw()	166
13.55 MessageQueue Class Reference	167
13.55.1 Detailed Description	167
13.55.2 Constructor & Destructor Documentation	167
13.55.2.1 MessageQueue()	167
13.55.2.2 ~MessageQueue()	167
13.55.3 Member Function Documentation	167
13.55.3.1 clear()	167
13.55.3.2 dequeue()	168
13.55.3.3 dequeueWait()	168
13.55.3.4 enqueue()	168
13.55.3.5 isEmpty()	168
13.55.3.6 size()	168
13.56 Model3D Struct Reference	169
13.56.1 Detailed Description	169
13.56.2 Field Documentation	169
13.56.2.1 boundingBoxMax	169
13.56.2.2 boundingBoxMin	169
13.56.2.3 modelId	169
13.56.2.4 scale	170
13.56.2.5 textureIds	170
13.57 ModelManager Class Reference	170
13.57.1 Detailed Description	171
13.57.2 Member Function Documentation	171
13.57.2.1 drawModel()	171
13.57.2.2 drawModelEx()	171
13.57.2.3 getBoundingBox()	172
13.57.2.4 getInstance()	172
13.57.2.5 getModelCount()	172
13.57.2.6 getModelId()	172

13.57.2.7 hasModel()	173
13.57.2.8 hasModelLibSet()	173
13.57.2.9 loadModel()	173
13.57.2.10 loadModelWithTextures()	174
13.57.2.11 setGraphicsLib()	175
13.57.2.12 unloadAllModels()	175
13.57.2.13 unloadModel()	175
13.58 Zappy::ModelManagerAdapter Class Reference	176
13.58.1 Detailed Description	176
13.58.2 Constructor & Destructor Documentation	176
13.58.2.1 ModelManagerAdapter()	176
13.58.3 Member Function Documentation	176
13.58.3.1 createShared()	176
13.58.3.2 getManager()	177
13.58.3.3 setGraphicsLib()	177
13.59 Zappy::ModelTileRenderStrategy Class Reference	177
13.59.1 Detailed Description	178
13.59.2 Constructor & Destructor Documentation	178
13.59.2.1 ModelTileRenderStrategy()	178
13.59.3 Member Function Documentation	178
13.59.3.1 renderTile()	178
13.60 raylibcpp::ModelWrap Class Reference	179
13.60.1 Detailed Description	179
13.60.2 Constructor & Destructor Documentation	179
13.60.2.1 ModelWrap()	179
13.60.2.2 ~ModelWrap()	180
13.60.3 Member Function Documentation	180
13.60.3.1 draw()	180
13.60.3.2 get()	180
13.61 raylibcpp::MusicWrap Class Reference	180
13.61.1 Detailed Description	180
13.61.2 Constructor & Destructor Documentation	180
13.61.2.1 MusicWrap()	181
13.61.2.2 ~MusicWrap()	181
13.61.3 Member Function Documentation	181
13.61.3.1 get()	181
13.61.3.2 pause()	181
13.61.3.3 play()	181
13.61.3.4 resume()	181
13.61.3.5 setVolume()	181
13.61.3.6 stop()	182
13.61.3.7 update()	182

13.62 NetworkLogger Class Reference	182
13.62.1 Detailed Description	182
13.62.2 Member Function Documentation	182
13.62.2.1 get()	182
13.62.2.2 isEnabled()	182
13.62.2.3 log()	183
13.62.2.4 setEnabled()	183
13.63 NetworkManager Class Reference	183
13.63.1 Detailed Description	183
13.63.2 Member Typedef Documentation	183
13.63.2.1 ConnectionCallback	183
13.63.2.2 MessageCallback	184
13.63.3 Constructor & Destructor Documentation	184
13.63.3.1 NetworkManager()	184
13.63.3.2 ~NetworkManager()	184
13.63.4 Member Function Documentation	184
13.63.4.1 connect()	184
13.63.4.2 disconnect()	185
13.63.4.3 isConnected()	185
13.63.4.4 networkThreadLoop()	185
13.63.4.5 processIncomingMessages()	185
13.63.4.6 sendCommand()	186
13.63.4.7 setConnectionCallback()	186
13.63.4.8 setMessageCallback()	186
13.64 NetworkThread Class Reference	186
13.64.1 Detailed Description	186
13.64.2 Constructor & Destructor Documentation	186
13.64.2.1 NetworkThread()	187
13.64.2.2 ~NetworkThread()	187
13.64.3 Member Function Documentation	187
13.64.3.1 isRunning()	187
13.64.3.2 start()	187
13.64.3.3 stop()	187
13.65 RayGUICPP::Panel Class Reference	187
13.65.1 Detailed Description	188
13.65.2 Member Function Documentation	188
13.65.2.1 Draw()	188
13.66 paringArgsClass.parseArgs Class Reference	188
13.66.1 Detailed Description	188
13.66.2 Constructor & Destructor Documentation	188
13.66.2.1 __init__()	189
13.66.2.2 parseArgs()	189

13.66.3 Member Function Documentation	189
13.66.3.1 getFlagValue()	189
13.66.3.2 getMachine()	189
13.66.3.3 getName()	190
13.66.3.4 getPort()	190
13.66.3.5 hasFlagAndValue()	190
13.66.3.6 searchUnknownFlag()	190
13.66.3.7 setMachine()	190
13.66.3.8 setName()	191
13.66.3.9 setPort()	191
13.66.4 Field Documentation	191
13.66.4.1 args	191
13.66.4.2 machine	191
13.66.4.3 name	191
13.66.4.4 parsed_args	191
13.66.4.5 port	191
13.67 parsing_info_s Struct Reference	192
13.67.1 Detailed Description	192
13.67.2 Field Documentation	192
13.67.2.1 client_nb	192
13.67.2.2 frequence	192
13.67.2.3 height	192
13.67.2.4 names	192
13.67.2.5 port	192
13.67.2.6 width	192
13.68 paringArgsClass.ParsingArgsError Class Reference	193
13.68.1 Detailed Description	193
13.68.2 Constructor & Destructor Documentation	193
13.68.2.1 __init__()	193
13.68.3 Field Documentation	194
13.68.3.1 message	194
13.69 ParsingCLI Class Reference	194
13.69.1 Detailed Description	194
13.69.2 Constructor & Destructor Documentation	194
13.69.2.1 ParsingCLI()	194
13.69.2.2 ~ParsingCLI()	194
13.69.3 Member Function Documentation	195
13.69.3.1 checkValidityCLI()	195
13.69.3.2 displayHelp()	195
13.69.3.3 getMachine()	195
13.69.3.4 getPort()	195
13.69.3.5 parse()	195

13.69.3.6 setMachine()	196
13.69.3.7 setPort()	196
13.70 pending_cmd_s Struct Reference	196
13.70.1 Detailed Description	196
13.70.2 Field Documentation	196
13.70.2.1 args	197
13.70.2.2 func	197
13.71 player_inventory_s Struct Reference	197
13.71.1 Detailed Description	197
13.71.2 Field Documentation	197
13.71.2.1 name	197
13.71.2.2 quantity	197
13.71.2.3 type	197
13.72 player_s Struct Reference	197
13.72.1 Detailed Description	198
13.72.2 Field Documentation	198
13.72.2.1 busy_until	198
13.72.2.2 command_queue	198
13.72.2.3 incantation_leader_id	199
13.72.2.4 inventory	199
13.72.2.5 inventory_size	199
13.72.2.6 is_in_incantation	199
13.72.2.7 level	199
13.72.2.8 life	199
13.72.2.9 pending_cmd	199
13.72.2.10 pos_x	199
13.72.2.11 pos_y	199
13.72.2.12 queue_size	199
13.72.2.13 rotation	200
13.72.2.14 team_name	200
13.73 PlayerInfoData Class Reference	200
13.73.1 Detailed Description	201
13.73.2 Constructor & Destructor Documentation	201
13.73.2.1 PlayerInfoData() [1/2]	201
13.73.2.2 PlayerInfoData() [2/2]	201
13.73.3 Member Function Documentation	201
13.73.3.1 getId()	201
13.73.3.2 getLevel()	201
13.73.3.3 getOrientation()	202
13.73.3.4 getTeamName()	202
13.73.3.5 getType()	202
13.73.3.6 getX()	202

13.73.3.7 getY()	202
13.73.3.8 setId()	202
13.73.3.9 setLevel()	202
13.73.3.10 setOrientation()	202
13.73.3.11 setTeamName()	203
13.73.3.12 setX()	203
13.73.3.13 setY()	203
13.74 PlayerInventoryData Class Reference	203
13.74.1 Detailed Description	204
13.74.2 Constructor & Destructor Documentation	204
13.74.2.1 PlayerInventoryData() [1/2]	204
13.74.2.2 PlayerInventoryData() [2/2]	205
13.74.3 Member Function Documentation	205
13.74.3.1 getDeraumere()	205
13.74.3.2 getFood()	205
13.74.3.3 getId()	205
13.74.3.4 getLinemate()	205
13.74.3.5 getMendiane()	205
13.74.3.6 getPhiras()	205
13.74.3.7 getSibur()	206
13.74.3.8 getThystame()	206
13.74.3.9 getType()	206
13.74.3.10 getX()	206
13.74.3.11 getY()	206
13.74.3.12 setDeraumere()	206
13.74.3.13 setFood()	206
13.74.3.14 setId()	206
13.74.3.15 setLinemate()	207
13.74.3.16 setMendiane()	207
13.74.3.17 setPhiras()	207
13.74.3.18 setSibur()	207
13.74.3.19 setThystame()	207
13.74.3.20 setX()	207
13.74.3.21 setY()	207
13.75 poll_manager_s Struct Reference	207
13.75.1 Detailed Description	208
13.75.2 Field Documentation	208
13.75.2.1 capacity	208
13.75.2.2 fds	208
13.75.2.3 needs_rebuild	208
13.76 RayGUICPP::ProgressBar Class Reference	208
13.76.1 Detailed Description	208

13.76.2 Member Function Documentation	209
13.76.2.1 Draw()	209
13.77 ProtocolParser Class Reference	209
13.77.1 Detailed Description	210
13.77.2 Member Typedef Documentation	210
13.77.2.1 ParseFunction	210
13.77.3 Constructor & Destructor Documentation	210
13.77.3.1 ProtocolParser()	210
13.77.3.2 ~ProtocolParser()	210
13.77.4 Member Function Documentation	211
13.77.4.1 extractCommandParameter()	211
13.77.4.2 extractMessageParameters()	211
13.77.4.3 getCommandName()	211
13.77.4.4 handleProtocol()	211
13.77.4.5 messageComplete()	211
13.77.4.6 parseAllTilesContent()	211
13.77.4.7 parseEggConnection()	212
13.77.4.8 parseEggDeath()	212
13.77.4.9 parseEggDrop()	212
13.77.4.10 parseEggLaying()	212
13.77.4.11 parseEndGame()	213
13.77.4.12 parseIncantationEnd()	213
13.77.4.13 parseIncantationStart()	213
13.77.4.14 parseIntParameter()	213
13.77.4.15 parseMapSize()	214
13.77.4.16 parseMessage() [1/2]	214
13.77.4.17 parseMessage() [2/2]	214
13.77.4.18 parsePlayerBroadcast()	214
13.77.4.19 parsePlayerConnection()	215
13.77.4.20 parsePlayerDeath()	215
13.77.4.21 parsePlayerExpulsion()	215
13.77.4.22 parsePlayerInventory()	215
13.77.4.23 parsePlayerLevel()	216
13.77.4.24 parsePlayerPosition()	216
13.77.4.25 parseRessourceCollect()	216
13.77.4.26 parseRessourceDrop()	216
13.77.4.27 parseServerMessage()	217
13.77.4.28 parseTeamNames()	217
13.77.4.29 parseTileContent()	217
13.77.4.30 parseTimeUnit()	217
13.77.4.31 parseUnknownCommand()	218
13.77.4.32 splitMessage()	218

13.78 RayGui Class Reference	218
13.78.1 Detailed Description	220
13.78.2 Constructor & Destructor Documentation	220
13.78.2.1 RayGui()	220
13.78.2.2 ~RayGui()	220
13.78.3 Member Function Documentation	220
13.78.3.1 ButtonPressed()	220
13.78.3.2 DrawButton()	220
13.78.3.3 DrawCheckBox()	221
13.78.3.4 DrawColorPicker()	221
13.78.3.5 DrawComboBox()	221
13.78.3.6 DrawDropdownBox()	221
13.78.3.7 DrawGroupBox()	222
13.78.3.8 DrawLabel()	222
13.78.3.9 DrawListView()	222
13.78.3.10 DrawMessageBox()	222
13.78.3.11 DrawPanel()	223
13.78.3.12 DrawProgressBar()	223
13.78.3.13 DrawScrollPane()	223
13.78.3.14 DrawSlider()	224
13.78.3.15 DrawSpinner()	224
13.78.3.16 DrawStatusBar()	224
13.78.3.17 DrawTabBar()	224
13.78.3.18 DrawTextBox()	225
13.78.3.19 DrawTextInputBox()	225
13.78.3.20 DrawToggle()	225
13.78.3.21 DrawToggleGroup()	226
13.78.3.22 DrawValueBox()	226
13.78.3.23 DrawWindowBox()	226
13.78.3.24 GetStyle()	226
13.78.3.25 IconText()	227
13.78.3.26 LoadStyle()	227
13.78.3.27 SetStyle()	227
13.79 RayGUICPP::RayGui Class Reference	227
13.79.1 Detailed Description	227
13.79.2 Constructor & Destructor Documentation	227
13.79.2.1 RayGui()	228
13.79.2.2 ~RayGui()	228
13.79.3 Member Function Documentation	228
13.79.3.1 GetStyle()	228
13.79.3.2 LoadStyle()	228
13.79.3.3 SetStyle()	228

13.80 RayLib Class Reference	228
13.80.1 Detailed Description	230
13.80.2 Constructor & Destructor Documentation	230
13.80.2.1 RayLib()	230
13.80.2.2 ~RayLib()	231
13.80.3 Member Function Documentation	231
13.80.3.1 BeginCamera3D()	231
13.80.3.2 BeginDrawing()	231
13.80.3.3 BindTexture3D()	231
13.80.3.4 ClearBackground()	231
13.80.3.5 CloseWindow()	231
13.80.3.6 CreateCamera2D()	232
13.80.3.7 CreateCamera3D()	232
13.80.3.8 DrawCircle()	232
13.80.3.9 DrawCube()	232
13.80.3.10 DrawCylinder()	233
13.80.3.11 DrawGrid()	233
13.80.3.12 DrawLine3D()	233
13.80.3.13 DrawModel3D()	233
13.80.3.14 DrawModelEx()	234
13.80.3.15 DrawPlane()	234
13.80.3.16 DrawRectangle()	234
13.80.3.17 DrawSphere()	234
13.80.3.18 DrawText()	235
13.80.3.19 DrawText3D()	235
13.80.3.20 DrawTexture2D()	235
13.80.3.21 DrawTextWave3D()	236
13.80.3.22 EndCamera3D()	236
13.80.3.23 EndDrawing()	236
13.80.3.24 GetMousePosition()	236
13.80.3.25 GetMouseWheelMove()	236
13.80.3.26 GetMouseX()	237
13.80.3.27 GetMouseY()	237
13.80.3.28 InitWindow()	237
13.80.3.29 IsKeyDown()	237
13.80.3.30 IsKeyPressed()	237
13.80.3.31 IsMouseButtonDown()	238
13.80.3.32 IsMouseButtonPressed()	238
13.80.3.33 IsMouseButtonReleased()	238
13.80.3.34 IsTextureReady()	238
13.80.3.35 LoadFont()	238
13.80.3.36 LoadModel3D()	238

13.80.3.37 LoadModelWithTexture()	239
13.80.3.38 LoadTexture2D()	239
13.80.3.39 LoadTexture3D()	239
13.80.3.40 MeasureText3D()	239
13.80.3.41 PlayMusic()	240
13.80.3.42 PlaySound()	240
13.80.3.43 setFps()	240
13.80.3.44 SetMusicVolume()	240
13.80.3.45 SetSoundVolume()	240
13.80.3.46 StopMusic()	241
13.80.3.47 StopSound()	241
13.80.3.48 UnloadFont()	241
13.80.3.49 UnloadModel3D()	241
13.80.3.50 UnloadTexture2D()	241
13.80.3.51 UnloadTexture3D()	241
13.80.3.52 UpdateMusic()	242
13.80.3.53 WindowShouldClose()	242
13.81 ZappyTypes::Rectangle Struct Reference	242
13.81.1 Detailed Description	242
13.81.2 Field Documentation	242
13.81.2.1 height	242
13.81.2.2 width	242
13.81.2.3 x	243
13.81.2.4 y	243
13.82 Renderer Class Reference	243
13.82.1 Detailed Description	243
13.82.2 Constructor & Destructor Documentation	243
13.82.2.1 Renderer()	243
13.82.2.2 ~Renderer()	243
13.82.3 Member Function Documentation	244
13.82.3.1 getMapHeight()	244
13.82.3.2 getMapWidth()	244
13.82.3.3 getResourceTextureId()	244
13.82.3.4 init()	244
13.82.3.5 loadResourceTexture()	244
13.82.3.6 render()	244
13.82.3.7 renderBackground()	245
13.82.3.8 renderGrid()	245
13.82.3.9 renderModelFromManager()	245
13.82.3.10 renderScene()	245
13.82.3.11 renderSprite2D()	245
13.82.3.12 renderUI()	246

13.83 raylibcpp::RenderState Struct Reference	246
13.83.1 Detailed Description	246
13.83.2 Field Documentation	246
13.83.2.1 charIndex	246
13.83.2.2 position	246
13.83.2.3 textOffsetX	246
13.83.2.4 textOffsetY	247
13.84 resource_dist_s Struct Reference	247
13.84.1 Detailed Description	247
13.84.2 Field Documentation	247
13.84.2.1 height	247
13.84.2.2 map	247
13.84.2.3 tile_indices	248
13.84.2.4 width	248
13.85 ResourceData Class Reference	248
13.85.1 Detailed Description	249
13.85.2 Constructor & Destructor Documentation	249
13.85.2.1 ResourceData()	249
13.85.3 Member Function Documentation	249
13.85.3.1 getPlayerId()	249
13.85.3.2 getResourceType()	249
13.85.3.3 getType()	249
13.85.3.4 setPlayerId()	249
13.85.3.5 setResourceType()	249
13.86 SystemWrapper::SafeBuffer Class Reference	250
13.86.1 Detailed Description	250
13.86.2 Constructor & Destructor Documentation	250
13.86.2.1 SafeBuffer()	250
13.86.2.2 ~SafeBuffer()	250
13.86.3 Member Function Documentation	250
13.86.3.1 data() [1/2]	250
13.86.3.2 data() [2/2]	250
13.86.3.3 resize()	251
13.86.3.4 setData()	251
13.86.3.5 size()	251
13.86.3.6 toString()	251
13.87 SystemWrapper::SafePollFd Class Reference	251
13.87.1 Detailed Description	252
13.87.2 Constructor & Destructor Documentation	252
13.87.2.1 SafePollFd() [1/2]	252
13.87.2.2 SafePollFd() [2/2]	252
13.87.3 Member Function Documentation	252

13.87.3.1 getEvents()	252
13.87.3.2 getFd()	252
13.87.3.3 getPollFd() [1/2]	252
13.87.3.4 getPollFd() [2/2]	252
13.87.3.5 getPollPtr() [1/2]	253
13.87.3.6 getPollPtr() [2/2]	253
13.87.3.7 getRevents()	253
13.87.3.8 setEvents()	253
13.87.3.9 setFd()	253
13.87.3.10 setRevents()	253
13.88 SystemWrapper::SafeSockAddr Class Reference	253
13.88.1 Detailed Description	254
13.88.2 Constructor & Destructor Documentation	254
13.88.2.1 SafeSockAddr() [1/3]	254
13.88.2.2 SafeSockAddr() [2/3]	254
13.88.2.3 SafeSockAddr() [3/3]	254
13.88.3 Member Function Documentation	254
13.88.3.1 getAddr() [1/2]	254
13.88.3.2 getAddr() [2/2]	254
13.88.3.3 getSize()	255
13.88.3.4 getSockAddr() [1/2]	255
13.88.3.5 getSockAddr() [2/2]	255
13.89 RayGUICPP::ScrollPane Class Reference	255
13.89.1 Detailed Description	255
13.89.2 Member Function Documentation	255
13.89.2.1 Draw()	255
13.90 server_s Struct Reference	256
13.90.1 Detailed Description	257
13.90.2 Field Documentation	257
13.90.2.1 client	257
13.90.2.2 current_resources	257
13.90.2.3 current_tick	257
13.90.2.4 eggs	257
13.90.2.5 graphical_clients	257
13.90.2.6 map	258
13.90.2.7 nfds	258
13.90.2.8 parsed_info	258
13.90.2.9 poll_manager	258
13.90.2.10 s_fd	258
13.90.2.11 serv_add	258
13.90.2.12 should_run	258
13.90.2.13 total_resources	258

13.91 ServerMessageData Class Reference	259
13.91.1 Detailed Description	259
13.91.2 Constructor & Destructor Documentation	259
13.91.2.1 ServerMessageData()	259
13.91.3 Member Function Documentation	260
13.91.3.1 getMessage()	260
13.91.3.2 getType()	260
13.91.3.3 setMessage()	260
13.92 raylibcpp::Shape Class Reference	260
13.92.1 Detailed Description	260
13.92.2 Member Function Documentation	260
13.92.2.1 drawCircle()	260
13.92.2.2 drawCube()	261
13.92.2.3 drawCylinder()	261
13.92.2.4 drawGrid()	261
13.92.2.5 drawLine3D()	261
13.92.2.6 drawPlane()	261
13.92.2.7 drawRectangle()	262
13.92.2.8 drawSphere()	262
13.93 Zappy::SimpleTileRenderStrategy Class Reference	262
13.93.1 Detailed Description	263
13.93.2 Constructor & Destructor Documentation	263
13.93.2.1 SimpleTileRenderStrategy()	263
13.93.3 Member Function Documentation	263
13.93.3.1 renderTile()	263
13.94 RayGUICPP::Slider Class Reference	264
13.94.1 Detailed Description	264
13.94.2 Member Function Documentation	264
13.94.2.1 Draw()	264
13.95 socketManager.SocketManager Class Reference	264
13.95.1 Detailed Description	265
13.95.2 Constructor & Destructor Documentation	265
13.95.2.1 __init__()	265
13.95.3 Member Function Documentation	265
13.95.3.1 get_message()	265
13.95.3.2 has_messages()	265
13.95.3.3 send_command()	266
13.95.3.4 start()	266
13.95.3.5 stop()	266
13.95.4 Field Documentation	266
13.95.4.1 buffer	266
13.95.4.2 lock	267

13.95.4.3 message_queue	267
13.95.4.4 pending_requests	267
13.95.4.5 running	267
13.95.4.6 sock	267
13.95.4.7 thread	267
13.96 raylibcpp::SoundWrap Class Reference	267
13.96.1 Detailed Description	267
13.96.2 Constructor & Destructor Documentation	268
13.96.2.1 SoundWrap()	268
13.96.2.2 ~SoundWrap()	268
13.96.3 Member Function Documentation	268
13.96.3.1 get()	268
13.96.3.2 pause()	268
13.96.3.3 play()	268
13.96.3.4 resume()	268
13.96.3.5 setVolume()	269
13.96.3.6 stop()	269
13.97 RayGUICPP::Spinner Class Reference	269
13.97.1 Detailed Description	269
13.97.2 Member Function Documentation	269
13.97.2.1 Draw()	269
13.98 RayGUICPP::StatusBar Class Reference	269
13.98.1 Detailed Description	270
13.98.2 Member Function Documentation	270
13.98.2.1 Draw()	270
13.99 RayGUICPP::TabBar Class Reference	270
13.99.1 Detailed Description	270
13.99.2 Member Function Documentation	270
13.99.2.1 Draw()	270
13.100 TcpConnection Class Reference	270
13.100.1 Detailed Description	271
13.100.2 Constructor & Destructor Documentation	271
13.100.2.1 TcpConnection()	271
13.100.2.2 ~TcpConnection()	271
13.100.3 Member Function Documentation	271
13.100.3.1 close()	271
13.100.3.2 connect()	271
13.100.3.3 hasData()	272
13.100.3.4 isConnected()	272
13.100.3.5 receive()	272
13.100.3.6 send()	272
13.101 TeamNameData Class Reference	273

13.101.1 Detailed Description	273
13.101.2 Constructor & Destructor Documentation	273
13.101.2.1 TeamNameData()	274
13.101.3 Member Function Documentation	274
13.101.3.1 addName()	274
13.101.3.2 getNames()	274
13.101.3.3 getType()	274
13.101.3.4 setNames()	274
13.102 raylibcpp::Text3D Class Reference	274
13.102.1 Detailed Description	275
13.102.2 Constructor & Destructor Documentation	275
13.102.2.1 Text3D()	275
13.102.2.2 ~Text3D()	275
13.102.3 Member Function Documentation	275
13.102.3.1 DrawText3D()	275
13.102.3.2 DrawTextCodepoint3D()	275
13.102.3.3 DrawTextWave3D()	276
13.102.3.4 MeasureTextWave3D()	276
13.103 raylibcpp::Text3DCodepoint Class Reference	276
13.103.1 Detailed Description	276
13.103.2 Member Function Documentation	276
13.103.2.1 draw()	276
13.104 raylibcpp::Text3DConfig Class Reference	277
13.104.1 Detailed Description	277
13.104.2 Member Function Documentation	277
13.104.2.1 adjustWaveIntensity()	277
13.104.2.2 adjustWaveSpeed()	278
13.104.2.3 createDefaultWaveConfig()	278
13.104.2.4 createFastWave()	278
13.104.2.5 createIntenseWave()	278
13.104.2.6 createSlowWave()	278
13.104.2.7 createSubtleWave()	278
13.104.2.8 createWaveConfig()	279
13.104.2.9 isShowLetterBoundary()	279
13.104.2.10 isShowTextBoundary()	279
13.104.2.11 isValidWaveConfig()	279
13.104.2.12 setShowLetterBoundary()	279
13.104.2.13 setShowTextBoundary()	280
13.105 raylibcpp::Text3DMeasurement Class Reference	280
13.105.1 Detailed Description	280
13.105.2 Member Function Documentation	280
13.105.2.1 measureText()	280

13.106 raylibcpp::Text3DParser Class Reference	281
13.106.1 Detailed Description	281
13.106.2 Member Function Documentation	281
13.106.2.1 isWaveMarker()	281
13.106.2.2 renderWaveText()	281
13.106.2.3 skipWaveMarkers()	282
13.107 raylibcpp::Text3DRenderer Class Reference	282
13.107.1 Detailed Description	283
13.107.2 Member Function Documentation	283
13.107.2.1 renderBasicText()	283
13.108 raylibcpp::Text3DWave Class Reference	283
13.108.1 Detailed Description	283
13.108.2 Member Function Documentation	284
13.108.2.1 applyWaveEffect()	284
13.108.2.2 createDefaultConfig()	284
13.108.2.3 isValidConfig()	284
13.109 RayGUICPP::TextBox Class Reference	284
13.109.1 Detailed Description	284
13.109.2 Member Function Documentation	285
13.109.2.1 Draw()	285
13.110 RayGUICPP::TextInputBox Class Reference	285
13.110.1 Detailed Description	285
13.110.2 Member Function Documentation	285
13.110.2.1 Draw()	285
13.111 raylibcpp::Texture Class Reference	286
13.111.1 Detailed Description	286
13.111.2 Constructor & Destructor Documentation	286
13.111.2.1 Texture()	286
13.111.2.2 ~Texture()	286
13.111.3 Member Function Documentation	286
13.111.3.1 draw()	286
13.111.3.2 get()	287
13.111.3.3 getNextId()	287
13.111.3.4 isReady()	287
13.112 raylibcpp::Texture3DWrap Class Reference	287
13.112.1 Detailed Description	287
13.112.2 Constructor & Destructor Documentation	287
13.112.2.1 Texture3DWrap()	287
13.112.2.2 ~Texture3DWrap()	288
13.112.3 Member Function Documentation	288
13.112.3.1 bind()	288
13.112.3.2 get()	288

13.113 raylibcpp::TextureCoordinates Struct Reference	288
13.113.1 Detailed Description	288
13.113.2 Field Documentation	288
13.113.2.1 th	288
13.113.2.2 tw	289
13.113.2.3 tx	289
13.113.2.4 ty	289
13.114 TextureManager Class Reference	289
13.114.1 Detailed Description	289
13.114.2 Member Function Documentation	290
13.114.2.1 drawTexture()	290
13.114.2.2 getInstance()	290
13.114.2.3 getTextureCount()	290
13.114.2.4 getTextureId()	291
13.114.2.5 hasTexture()	291
13.114.2.6 hasTextureLibSet()	291
13.114.2.7 loadTexture()	291
13.114.2.8 setGraphicsLib()	292
13.114.2.9 unloadAllTextures()	292
13.114.2.10 unloadTexture()	293
13.115 tile_s Struct Reference	293
13.115.1 Detailed Description	293
13.115.2 Field Documentation	293
13.115.2.1 egg_capacity	293
13.115.2.2 egg_count	293
13.115.2.3 egg_ids	294
13.115.2.4 player_capacity	294
13.115.2.5 player_count	294
13.115.2.6 player_ids	294
13.115.2.7 resources	294
13.116 TileContentData Class Reference	294
13.116.1 Detailed Description	295
13.116.2 Constructor & Destructor Documentation	295
13.116.2.1 TileContentData()	295
13.116.3 Member Function Documentation	296
13.116.3.1 getDeraumere()	296
13.116.3.2 getFood()	296
13.116.3.3 getLinemate()	296
13.116.3.4 getMendiane()	296
13.116.3.5 getPhiras()	296
13.116.3.6 getSibur()	296
13.116.3.7 getThystame()	296

13.116.3.8 getType()	297
13.116.3.9 getX()	297
13.116.3.10 getY()	297
13.116.3.11 setDeraumere()	297
13.116.3.12 setFood()	297
13.116.3.13 setLinemate()	297
13.116.3.14 setMendiane()	297
13.116.3.15 setPhiras()	297
13.116.3.16 setSibur()	298
13.116.3.17 setThystame()	298
13.116.3.18 setX()	298
13.116.3.19 setY()	298
13.117 TileData Struct Reference	298
13.117.1 Detailed Description	299
13.117.2 Field Documentation	299
13.117.2.1 eggIds	299
13.117.2.2 isIncantating	299
13.117.2.3 playerIds	299
13.117.2.4 resources	299
13.118 Zappy::TileRenderStrategyFactory Class Reference	299
13.118.1 Detailed Description	299
13.118.2 Constructor & Destructor Documentation	300
13.118.2.1 TileRenderStrategyFactory()	300
13.118.3 Member Function Documentation	300
13.118.3.1 createDetailedTileStrategy()	300
13.118.3.2 createModelTileStrategy()	300
13.118.3.3 createSimpleTileStrategy()	300
13.119 TimeUnitData Class Reference	300
13.119.1 Detailed Description	301
13.119.2 Constructor & Destructor Documentation	301
13.119.2.1 TimeUnitData()	301
13.119.3 Member Function Documentation	301
13.119.3.1 getTimeUnit()	302
13.119.3.2 getType()	302
13.119.3.3 setTimeUnit()	302
13.120 RayGUICPP::Toggle Class Reference	302
13.120.1 Detailed Description	302
13.120.2 Member Function Documentation	302
13.120.2.1 Draw()	302
13.121 RayGUICPP::ToggleGroup Class Reference	302
13.121.1 Detailed Description	303
13.121.2 Member Function Documentation	303

13.121.2.1 Draw()	303
13.122 UIRenderer Class Reference	303
13.122.1 Detailed Description	303
13.122.2 Constructor & Destructor Documentation	303
13.122.2.1 UIRenderer()	303
13.122.2.2 ~UIRenderer()	304
13.122.3 Member Function Documentation	304
13.122.3.1 fps()	304
13.122.3.2 renderCameralInfo()	304
13.122.3.3 renderFPS()	304
13.122.3.4 renderStatusBar()	304
13.122.3.5 renderUI()	304
13.122.3.6 showUI()	305
13.123 behaviors.UpgradeBehavior Class Reference	305
13.123.1 Detailed Description	306
13.123.2 Member Function Documentation	306
13.123.2.1 execute()	306
13.124 raylibcpp::Utils Class Reference	306
13.124.1 Detailed Description	306
13.124.2 Member Function Documentation	307
13.124.2.1 hexToColor()	307
13.125 RayGUICPP::ValueBox Class Reference	307
13.125.1 Detailed Description	307
13.125.2 Member Function Documentation	307
13.125.2.1 Draw()	307
13.126 ZappyTypes::Vector2 Struct Reference	307
13.126.1 Detailed Description	308
13.126.2 Field Documentation	308
13.126.2.1 x	308
13.126.2.2 y	308
13.127 ZappyTypes::Vector3 Struct Reference	308
13.127.1 Detailed Description	308
13.127.2 Field Documentation	308
13.127.2.1 x	308
13.127.2.2 y	308
13.127.2.3 z	309
13.128 raylibcpp::Vector3Wrap Class Reference	309
13.128.1 Detailed Description	309
13.128.2 Constructor & Destructor Documentation	309
13.128.2.1 Vector3Wrap()	309
13.128.3 Member Function Documentation	309
13.128.3.1 add()	309

13.128.3.2 cross()	310
13.128.3.3 get()	310
13.128.3.4 normalize()	310
13.128.3.5 scale()	310
13.128.3.6 set()	310
13.128.3.7 sub()	310
13.128.3.8 transform()	311
13.129 raylibcpp::WaveTextConfig Struct Reference	311
13.129.1 Detailed Description	311
13.129.2 Field Documentation	311
13.129.2.1 waveOffset	311
13.129.2.2 waveRange	311
13.129.2.3 waveSpeed	311
13.130 raylibcpp::Window Class Reference	311
13.130.1 Detailed Description	312
13.130.2 Constructor & Destructor Documentation	312
13.130.2.1 Window()	312
13.130.2.2 ~Window()	312
13.130.3 Member Function Documentation	312
13.130.3.1 beginDrawing()	312
13.130.3.2 clear()	312
13.130.3.3 endDrawing()	313
13.130.3.4 setFps()	313
13.130.3.5 shouldClose()	313
13.131 RayGUICPP::WindowBox Class Reference	313
13.131.1 Detailed Description	313
13.131.2 Member Function Documentation	313
13.131.2.1 Draw()	313
14 File Documentation	315
14.1 /root/Desktop/Zappy/docs/index.md File Reference	315
14.2 /root/Desktop/Zappy/docs/protocol_server_ai.md File Reference	315
14.3 /root/Desktop/Zappy/libs/RayGUICPP/include/RayGuiEncap.hpp File Reference	315
14.4 /root/Desktop/Zappy/libs/RayGUICPP/RayGui.cpp File Reference	317
14.4.1 Function Documentation	317
14.4.1.1 createGuiLib()	317
14.5 /root/Desktop/Zappy/libs/RayGUICPP/RayGui.hpp File Reference	317
14.6 /root/Desktop/Zappy/libs/RayGUICPP/raygui_impl.cpp File Reference	318
14.6.1 Macro Definition Documentation	319
14.6.1.1 RAYGUI_IMPLEMENTATION	319
14.6.1.2 RAYLIB_COLOR_DEFINED	319
14.6.1.3 RAYLIB_RECTANGLE_DEFINED	319

14.6.1.4 RAYLIB_VECTOR2_DEFINED	320
14.6.1.5 RAYLIB_VECTOR3_DEFINED	320
14.7 /root/Desktop/Zappy/libs/RaylibCPP/audio/Audio.cpp File Reference	320
14.8 /root/Desktop/Zappy/libs/RaylibCPP/audio/Audio.hpp File Reference	320
14.9 /root/Desktop/Zappy/libs/RaylibCPP/camera/Camera.cpp File Reference	322
14.10 /root/Desktop/Zappy/libs/RaylibCPP/camera/Camera.hpp File Reference	322
14.11 /root/Desktop/Zappy/libs/RaylibCPP/font/Font.cpp File Reference	323
14.12 /root/Desktop/Zappy/libs/RaylibCPP/font/Font.hpp File Reference	324
14.13 /root/Desktop/Zappy/libs/RaylibCPP/font/Text3D.cpp File Reference	325
14.14 /root/Desktop/Zappy/libs/RaylibCPP/font/Text3D.hpp File Reference	326
14.15 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DCodepoint.cpp File Reference	327
14.16 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DCodepoint.hpp File Reference	327
14.17 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DConfig.cpp File Reference	328
14.18 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DConfig.hpp File Reference	329
14.19 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DCore.hpp File Reference	330
14.19.1 Macro Definition Documentation	331
14.19.1.1 LETTER_BOUNDARY_COLOR	331
14.19.1.2 LETTER_BOUNDARY_SIZE	332
14.19.1.3 TEXT_MAX_LAYERS	332
14.20 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DHelper.cpp File Reference	332
14.21 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DHelper.hpp File Reference	333
14.22 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DMeasurement.cpp File Reference	334
14.23 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DMeasurement.hpp File Reference	335
14.24 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DParser.cpp File Reference	335
14.25 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DParser.hpp File Reference	336
14.26 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DRenderer.cpp File Reference	337
14.27 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DRenderer.hpp File Reference	338
14.28 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DWave.cpp File Reference	339
14.29 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DWave.hpp File Reference	339
14.30 /root/Desktop/Zappy/libs/RaylibCPP/input/Input.cpp File Reference	340
14.31 /root/Desktop/Zappy/libs/RaylibCPP/input/Input.hpp File Reference	341
14.32 /root/Desktop/Zappy/libs/RaylibCPP/RayLib.cpp File Reference	342
14.32.1 Function Documentation	343
14.32.1.1 createGraphicsLib()	343
14.33 /root/Desktop/Zappy/libs/RaylibCPP/RayLib.hpp File Reference	343
14.34 /root/Desktop/Zappy/libs/RaylibCPP/shape/Shape.cpp File Reference	344
14.35 /root/Desktop/Zappy/libs/RaylibCPP/shape/Shape.hpp File Reference	344
14.36 /root/Desktop/Zappy/libs/RaylibCPP/texture/Texture.cpp File Reference	345
14.37 /root/Desktop/Zappy/libs/RaylibCPP/texture/Texture.hpp File Reference	346
14.38 /root/Desktop/Zappy/libs/RaylibCPP/utils/Utils.cpp File Reference	347
14.39 /root/Desktop/Zappy/libs/RaylibCPP/utils/Utils.hpp File Reference	348
14.40 /root/Desktop/Zappy/libs/RaylibCPP/window/Window.cpp File Reference	349

14.41 /root/Desktop/Zappy/libs/RaylibCPP/window/Window.hpp File Reference	350
14.42 /root/Desktop/Zappy/libs/TypeAdapter.hpp File Reference	351
14.43 /root/Desktop/Zappy/src/AI/agent/actions.py File Reference	352
14.44 /root/Desktop/Zappy/src/AI/agent/agent.py File Reference	352
14.45 /root/Desktop/Zappy/src/AI/agent/behaviors.py File Reference	352
14.46 /root/Desktop/Zappy/src/AI/agent/broadcastManager.py File Reference	352
14.47 /root/Desktop/Zappy/src/AI/agent/decisionManager.py File Reference	352
14.48 /root/Desktop/Zappy/src/AI/agent/socketManager.py File Reference	353
14.49 /root/Desktop/Zappy/src/AI/constants/resources.py File Reference	353
14.50 /root/Desktop/Zappy/src/AI/constants/roles.py File Reference	353
14.51 /root/Desktop/Zappy/src/AI/constants/upgrades.py File Reference	353
14.52 /root/Desktop/Zappy/src/AI/logger/logger.py File Reference	353
14.53 /root/Desktop/Zappy/src/AI/main.py File Reference	353
14.54 /root/Desktop/Zappy/src/AI/parser/paringArgsClass.py File Reference	354
14.55 /root/Desktop/Zappy/src/AI/requirements.txt File Reference	354
14.56 /root/Desktop/Zappy/src/AI/utils/encryption.py File Reference	354
14.57 /root/Desktop/Zappy/src/AI/utils/input.py File Reference	354
14.58 /root/Desktop/Zappy/src/AI/utils/zappy.py File Reference	355
14.59 /root/Desktop/Zappy/src/GUI/cameraController/CameraController.cpp File Reference	355
14.60 /root/Desktop/Zappy/src/GUI/cameraController/CameraController.hpp File Reference	355
14.61 /root/Desktop/Zappy/src/GUI/Constants.hpp File Reference	356
14.61.1 Enumeration Type Documentation	358
14.61.1.1 KeyboardKey	358
14.61.2 Variable Documentation	359
14.61.2.1 DEFAULT_HEIGHT	359
14.61.2.2 DEFAULT_TITLE	360
14.61.2.3 DEFAULT_WIDTH	360
14.61.2.4 NUM_RANDOM_MODELS	360
14.62 /root/Desktop/Zappy/src/GUI/GameLoop.cpp File Reference	360
14.63 /root/Desktop/Zappy/src/GUI/GameLoop.hpp File Reference	360
14.64 /root/Desktop/Zappy/src/GUI/graphicalContext/GraphicalContext.cpp File Reference	361
14.65 /root/Desktop/Zappy/src/GUI/graphicalContext/GraphicalContext.hpp File Reference	361
14.65.1 Enumeration Type Documentation	362
14.65.1.1 ResourceType	362
14.66 /root/Desktop/Zappy/src/GUI/main.cpp File Reference	363
14.66.1 Function Documentation	363
14.66.1.1 displayHelp()	363
14.66.1.2 main()	363
14.67 /root/Desktop/Zappy/src/GUI/network/buffer/MessageQueue.cpp File Reference	364
14.68 /root/Desktop/Zappy/src/GUI/network/buffer/MessageQueue.hpp File Reference	364
14.69 /root/Desktop/Zappy/src/GUI/network/connection/TcpConnection.cpp File Reference	365
14.70 /root/Desktop/Zappy/src/GUI/network/connection/TcpConnection.hpp File Reference	366

14.71 /root/Desktop/Zappy/src/GUI/network/networkManager/NetworkLogger.hpp File Reference	367
14.72 /root/Desktop/Zappy/src/GUI/network/networkManager/NetworkManager.cpp File Reference	367
14.73 /root/Desktop/Zappy/src/GUI/network/networkManager/NetworkManager.hpp File Reference	368
14.74 /root/Desktop/Zappy/src/GUI/network/networkThread/NetworkThread.cpp File Reference	369
14.75 /root/Desktop/Zappy/src/GUI/network/networkThread/NetworkThread.hpp File Reference	369
14.76 /root/Desktop/Zappy/src/GUI/network/protocol/HeaderMessage.hpp File Reference	370
14.76.1 Macro Definition Documentation	371
14.76.1.1 BCT_HEADER	371
14.76.1.2 EBO_HEADER	371
14.76.1.3 EDI_HEADER	371
14.76.1.4 ENW_HEADER	371
14.76.1.5 MCT_HEADER	371
14.76.1.6 MSZ_HEADER	371
14.76.1.7 PBC_HEADER	372
14.76.1.8 PDI_HEADER	372
14.76.1.9 PDR_HEADER	372
14.76.1.10 PEX_HEADER	372
14.76.1.11 PFK_HEADER	372
14.76.1.12 PGT_HEADER	372
14.76.1.13 PIC_HEADER	372
14.76.1.14 PIE_HEADER	372
14.76.1.15 PIN_HEADER	372
14.76.1.16 PLV_HEADER	372
14.76.1.17 PNW_HEADER	373
14.76.1.18 PPO_HEADER	373
14.76.1.19 SBP_HEADER	373
14.76.1.20 SEG_HEADER	373
14.76.1.21 SGT_HEADER	373
14.76.1.22 SMG_HEADER	373
14.76.1.23 SST_HEADER	373
14.76.1.24 SUC_HEADER	373
14.76.1.25 TNA_HEADER	373
14.77 /root/Desktop/Zappy/src/GUI/network/protocol/Message.cpp File Reference	373
14.78 /root/Desktop/Zappy/src/GUI/network/protocol/Message.hpp File Reference	374
14.79 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/BroadcastData.hpp File Reference	374
14.80 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/EggData.hpp File Reference	375
14.81 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/EndGameData.hpp File Reference	376
14.82 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/IMessageData.hpp File Reference	377
14.82.1 Enumeration Type Documentation	378
14.82.1.1 MessageType	378
14.83 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/IncantationData.hpp File Reference	378
14.84 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/IncantationEndData.hpp File Reference	379

14.85 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/MapSizeData.hpp File Reference . . .	380
14.86 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/MessageDataAll.hpp File Reference . . .	381
14.87 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/PlayerInfoData.hpp File Reference . . .	382
14.88 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/PlayerInventoryData.hpp File Reference . . .	383
14.89 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/ResourceData.hpp File Reference . . .	384
14.90 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/ServerMessageData.hpp File Reference . . .	385
14.91 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/TeamNameData.hpp File Reference . . .	386
14.92 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/TileContentData.hpp File Reference . . .	387
14.93 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/TimeUnitData.hpp File Reference . . .	388
14.94 /root/Desktop/Zappy/src/GUI/network/protocol/ProtocolParser.cpp File Reference	389
14.95 /root/Desktop/Zappy/src/GUI/network/protocol/ProtocolParser.hpp File Reference	389
14.96 /root/Desktop/Zappy/src/GUI/network/utils/SystemWrapper.cpp File Reference	390
14.97 /root/Desktop/Zappy/src/GUI/network/utils/SystemWrapper.hpp File Reference	391
14.98 /root/Desktop/Zappy/src/parsing/ParsingCLI.cpp File Reference	393
14.99 /root/Desktop/Zappy/src/parsing/ParsingCLI.hpp File Reference	393
14.100 /root/Desktop/Zappy/src/GUI/README.md File Reference	395
14.101 /root/Desktop/Zappy/src/Server/README.md File Reference	395
14.102 /root/Desktop/Zappy/README.md File Reference	395
14.103 /root/Desktop/Zappy/libs/RayGUICPP/README.md File Reference	395
14.104 /root/Desktop/Zappy/libs/RaylibCPP/README.md File Reference	395
14.105 /root/Desktop/Zappy/src/GUI/renderer/MapRenderer.cpp File Reference	395
14.106 /root/Desktop/Zappy/src/GUI/renderer/MapRenderer.hpp File Reference	395
14.107 /root/Desktop/Zappy/src/GUI/renderer/MapRendererObserver.cpp File Reference	396
14.108 /root/Desktop/Zappy/src/GUI/renderer/MapRendererObserver.hpp File Reference	397
14.109 /root/Desktop/Zappy/src/GUI/renderer/Renderer.cpp File Reference	397
14.110 /root/Desktop/Zappy/src/GUI/renderer/Renderer.hpp File Reference	398
14.111 /root/Desktop/Zappy/src/GUI/renderer/strategies/DetailedTileRenderStrategy.cpp File Reference . . .	399
14.112 /root/Desktop/Zappy/src/GUI/renderer/strategies/DetailedTileRenderStrategy.hpp File Reference . . .	399
14.113 /root/Desktop/Zappy/src/GUI/renderer/strategies/ITileRenderStrategy.hpp File Reference	400
14.114 /root/Desktop/Zappy/src/GUI/renderer/strategies/ModelTileRenderStrategy.cpp File Reference . . .	401
14.115 /root/Desktop/Zappy/src/GUI/renderer/strategies/ModelTileRenderStrategy.hpp File Reference . . .	401
14.116 /root/Desktop/Zappy/src/GUI/renderer/strategies/SimpleTileRenderStrategy.cpp File Reference . . .	402
14.117 /root/Desktop/Zappy/src/GUI/renderer/strategies/SimpleTileRenderStrategy.hpp File Reference . . .	403
14.118 /root/Desktop/Zappy/src/GUI/renderer/strategies/TileRenderStrategyFactory.cpp File Reference . . .	404
14.119 /root/Desktop/Zappy/src/GUI/renderer/strategies/TileRenderStrategyFactory.hpp File Reference . . .	404
14.120 /root/Desktop/Zappy/src/GUI/renderer/UIRenderer.cpp File Reference	405
14.121 /root/Desktop/Zappy/src/GUI/renderer/UIRenderer.hpp File Reference	406
14.122 /root/Desktop/Zappy/src/GUI/shared/exception/AException.cpp File Reference	407
14.123 /root/Desktop/Zappy/src/GUI/shared/exception/AException.hpp File Reference	408
14.124 /root/Desktop/Zappy/src/GUI/shared/exception/IException.hpp File Reference	408
14.125 /root/Desktop/Zappy/src/GUI/textureManager/ModelManager.cpp File Reference	409

14.126 /root/Desktop/Zappy/src/GUI/textureManager/ModelManager.hpp File Reference	410
14.127 /root/Desktop/Zappy/src/GUI/textureManager/ModelManagerAdapter.hpp File Reference	411
14.128 /root/Desktop/Zappy/src/GUI/textureManager/TextureManager.cpp File Reference	412
14.129 /root/Desktop/Zappy/src/GUI/textureManager/TextureManager.hpp File Reference	413
14.130 /root/Desktop/Zappy/src/Server/buffer/circular_buffer.c File Reference	414
14.130.1 Function Documentation	415
14.130.1.1 add_to_circular_buffer()	415
14.130.1.2 extract_command()	415
14.130.1.3 find_command_end()	415
14.130.1.4 init_circular_buffer()	415
14.131 /root/Desktop/Zappy/src/Server/command/broadcast.c File Reference	416
14.131.1 Function Documentation	416
14.131.1.1 broadcast()	416
14.131.1.2 build_broadcast_message()	417
14.131.1.3 calculate_direction()	417
14.131.1.4 calculate_message_length()	417
14.131.1.5 calculate_shortest_distance_x()	418
14.131.1.6 calculate_shortest_distance_y()	418
14.131.1.7 concatenate_message_parts()	418
14.131.1.8 get_base_direction()	418
14.131.1.9 send_broadcast_to_client()	419
14.132 /root/Desktop/Zappy/src/Server/command/connect_nbr.c File Reference	419
14.132.1 Function Documentation	420
14.132.1.1 connect_nbr()	420
14.132.1.2 connect_nbr_srv()	420
14.132.1.3 count_team_eggs()	421
14.132.1.4 count_team_players()	421
14.132.1.5 format_response()	421
14.133 /root/Desktop/Zappy/src/Server/command/eject.c File Reference	421
14.133.1 Function Documentation	422
14.133.1.1 convert_rotation_to_vector()	422
14.133.1.2 direction_push()	423
14.133.1.3 eject()	423
14.133.1.4 get_string_to_send()	423
14.133.1.5 push_client()	423
14.133.1.6 push_eggs()	424
14.133.1.7 push_single_client()	424
14.133.1.8 wrap_position()	424
14.134 /root/Desktop/Zappy/src/Server/command/finish_incantation.c File Reference	425
14.134.1 Function Documentation	425
14.134.1.1 check_win()	425
14.134.1.2 finish_incantation()	426

14.134.1.3 handle_incantation_failure()	426
14.134.1.4 handle_incantation_success()	426
14.134.1.5 handle_win()	427
14.134.1.6 remove_resources()	427
14.135 /root/Desktop/Zappy/src/Server/command/fork.c File Reference	427
14.135.1 Function Documentation	428
14.135.1.1 fork_c()	428
14.135.1.2 get_next_egg_id()	428
14.136 /root/Desktop/Zappy/src/Server/command/forward.c File Reference	429
14.136.1 Function Documentation	429
14.136.1.1 change_map_pos()	429
14.136.1.2 forward()	430
14.136.1.3 wrap_position()	430
14.137 /root/Desktop/Zappy/src/Server/command/inventory.c File Reference	430
14.137.1 Function Documentation	431
14.137.1.1 inventory()	431
14.138 /root/Desktop/Zappy/src/Server/command/left.c File Reference	431
14.138.1 Function Documentation	432
14.138.1.1 change_rot()	432
14.138.1.2 left()	432
14.139 /root/Desktop/Zappy/src/Server/command/look.c File Reference	433
14.139.1 Function Documentation	433
14.139.1.1 format_look()	433
14.139.1.2 format_tile()	434
14.139.1.3 get_total_size()	434
14.139.1.4 get_total_size_tile()	434
14.139.1.5 look()	435
14.139.1.6 look_tiles()	435
14.140 /root/Desktop/Zappy/src/Server/command/look_utils.c File Reference	435
14.140.1 Function Documentation	436
14.140.1.1 check_down()	436
14.140.1.2 check_left()	436
14.140.1.3 check_right()	437
14.140.1.4 check_rota_tiles()	437
14.140.1.5 check_up()	437
14.141 /root/Desktop/Zappy/src/Server/command/parse_command.c File Reference	438
14.141.1 Function Documentation	438
14.141.1.1 check_command()	438
14.141.1.2 check_disconnect()	439
14.141.1.3 execute_com()	439
14.141.1.4 execute_graphical_command()	439
14.141.1.5 execute_if_free()	439

14.141.1.6 find_and_execute()	440
14.141.1.7 get_command_data()	440
14.141.1.8 get_message()	440
14.141.1.9 send_info_new_client()	441
14.142 /root/Desktop/Zappy/src/Server/command/parse_command_utils.c File Reference	441
14.142.1 Function Documentation	442
14.142.1.1 add_pending_cmd()	442
14.142.1.2 cleanup_client()	442
14.142.1.3 cleanup_pending()	443
14.142.1.4 cleanup_player_queue()	443
14.142.1.5 write_command_output()	443
14.143 /root/Desktop/Zappy/src/Server/command/parse_command_utils_bis.c File Reference	443
14.143.1 Function Documentation	444
14.143.1.1 can_connect()	444
14.143.1.2 is_valid_team_name()	444
14.144 /root/Desktop/Zappy/src/Server/command/pending_cmd_utils.c File Reference	445
14.144.1 Function Documentation	445
14.144.1.1 execute_pending_cmd()	446
14.145 /root/Desktop/Zappy/src/Server/command/right.c File Reference	446
14.145.1 Function Documentation	446
14.145.1.1 change_rot()	447
14.145.1.2 right()	447
14.146 /root/Desktop/Zappy/src/Server/command/set_object.c File Reference	447
14.146.1 Function Documentation	448
14.146.1.1 set_object()	448
14.147 /root/Desktop/Zappy/src/Server/command/start_incantation.c File Reference	448
14.147.1 Function Documentation	449
14.147.1.1 can_start_incantation()	449
14.147.1.2 get_client_by_id_player()	450
14.147.1.3 has_enough_resources()	450
14.147.1.4 how_many_players_needed()	450
14.147.1.5 nb_valid_players_on_tile()	450
14.147.1.6 set_busy_all()	451
14.147.1.7 set_busy_until()	451
14.147.1.8 start_incantation()	451
14.148 /root/Desktop/Zappy/src/Server/command/take_object.c File Reference	451
14.148.1 Function Documentation	452
14.148.1.1 determine_type()	452
14.148.1.2 take_object()	452
14.148.1.3 update_resources()	453
14.149 /root/Desktop/Zappy/src/Server/command/tile_to_str.c File Reference	453
14.149.1 Function Documentation	454

14.149.1.1 add_resources()	454
14.149.1.2 calculate_total_size()	454
14.149.1.3 format_str_resources()	454
14.149.1.4 repeat_word()	455
14.149.1.5 tile_to_str()	455
14.150 /root/Desktop/Zappy/src/Server/free.c File Reference	455
14.150.1 Function Documentation	456
14.150.1.1 free_all()	456
14.150.1.2 free_arr()	457
14.150.1.3 free_client()	457
14.150.1.4 free_eggs()	457
14.150.1.5 free_map()	457
14.150.1.6 free_parsed_info()	458
14.150.1.7 free_poll_mana()	458
14.150.1.8 free_tiles()	458
14.151 /root/Desktop/Zappy/src/Server/graphical_command/command_bct.c File Reference	458
14.151.1 Function Documentation	459
14.151.1.1 calculate_size_bct_command()	459
14.151.1.2 command_bct()	459
14.151.1.3 command_mtc()	460
14.151.1.4 get_buffer_bct_command()	460
14.151.1.5 send_bct_command()	460
14.151.1.6 send_bct_to_all_graphical_clients()	461
14.151.1.7 send_mtc_to_all_graphical_clients()	461
14.151.1.8 send_tile_content_to_one_client()	461
14.152 /root/Desktop/Zappy/src/Server/graphical_command/command_ebo.c File Reference	461
14.152.1 Function Documentation	462
14.152.1.1 send_ebo_command()	462
14.153 /root/Desktop/Zappy/src/Server/graphical_command/command_edi.c File Reference	462
14.153.1 Function Documentation	463
14.153.1.1 send_edi_command()	463
14.154 /root/Desktop/Zappy/src/Server/graphical_command/command_enw.c File Reference	464
14.154.1 Function Documentation	464
14.154.1.1 get_buffer_for_enw()	464
14.154.1.2 send_enw_command()	465
14.154.1.3 send_enw_command_start()	465
14.154.1.4 send_enw_command_to_client()	465
14.155 /root/Desktop/Zappy/src/Server/graphical_command/command_msz.c File Reference	466
14.155.1 Function Documentation	466
14.155.1.1 command_msz()	466
14.155.1.2 send_msz_command()	466
14.156 /root/Desktop/Zappy/src/Server/graphical_command/command_pbc.c File Reference	467

14.156.1 Function Documentation	467
14.156.1.1 command_pbc()	467
14.157 /root/Desktop/Zappy/src/Server/graphical_command/command_pdi.c File Reference	468
14.157.1 Function Documentation	468
14.157.1.1 command_pdi()	469
14.158 /root/Desktop/Zappy/src/Server/graphical_command/command_pdr.c File Reference	469
14.158.1 Function Documentation	469
14.158.1.1 command_pdr()	470
14.159 /root/Desktop/Zappy/src/Server/graphical_command/command_pex.c File Reference	470
14.159.1 Function Documentation	470
14.159.1.1 command_pex()	471
14.160 /root/Desktop/Zappy/src/Server/graphical_command/command_pkf.c File Reference	471
14.160.1 Function Documentation	471
14.160.1.1 command_pkf()	472
14.161 /root/Desktop/Zappy/src/Server/graphical_command/command_pgt.c File Reference	472
14.161.1 Function Documentation	472
14.161.1.1 command_pgt()	473
14.162 /root/Desktop/Zappy/src/Server/graphical_command/command_pic.c File Reference	473
14.162.1 Function Documentation	474
14.162.1.1 command_pic()	474
14.162.1.2 get_pic_buffer()	474
14.163 /root/Desktop/Zappy/src/Server/graphical_command/command_pie.c File Reference	474
14.163.1 Function Documentation	475
14.163.1.1 command_pie()	475
14.163.1.2 get_pie_buffer()	476
14.164 /root/Desktop/Zappy/src/Server/graphical_command/command_pin.c File Reference	476
14.164.1 Function Documentation	476
14.164.1.1 calculate_size_pin_command()	477
14.164.1.2 command_pin()	477
14.164.1.3 get_buffer_pin_command()	477
14.164.1.4 send_pin_command()	477
14.164.1.5 send_pin_to_all()	478
14.165 /root/Desktop/Zappy/src/Server/graphical_command/command_plv.c File Reference	478
14.165.1 Function Documentation	479
14.165.1.1 command_plv()	479
14.165.1.2 send_plv_command()	479
14.165.1.3 send_plv_to_all()	479
14.166 /root/Desktop/Zappy/src/Server/graphical_command/command_pnw.c File Reference	480
14.166.1 Function Documentation	480
14.166.1.1 calcul_size_pnw_command()	480
14.166.1.2 send_all_player_info_to_one_client()	481
14.166.1.3 send_pnw_command()	481

14.166.1.4 send_pnw_command_to_all()	481
14.167 /root/Desktop/Zappy/src/Server/graphical_command/command_ppo.c File Reference	482
14.167.1 Function Documentation	482
14.167.1.1 command_ppo()	482
14.167.1.2 find_client_by_id()	483
14.167.1.3 get_ppo_buffer()	483
14.167.1.4 send_ppo_command()	483
14.168 /root/Desktop/Zappy/src/Server/graphical_command/command_seg.c File Reference	483
14.168.1 Function Documentation	484
14.168.1.1 command_seg()	484
14.169 /root/Desktop/Zappy/src/Server/graphical_command/command_sgt.c File Reference	485
14.169.1 Function Documentation	485
14.169.1.1 command_sgt()	485
14.170 /root/Desktop/Zappy/src/Server/graphical_command/command_smg.c File Reference	486
14.170.1 Function Documentation	486
14.170.1.1 send_smg_command()	486
14.171 /root/Desktop/Zappy/src/Server/graphical_command/command_sst.c File Reference	487
14.171.1 Function Documentation	487
14.171.1.1 command_sst()	487
14.171.1.2 get_buffer_sst()	488
14.171.1.3 get_time_from_buffer()	488
14.172 /root/Desktop/Zappy/src/Server/graphical_command/command_tna.c File Reference	488
14.172.1 Function Documentation	489
14.172.1.1 command_tna()	489
14.172.1.2 send_one_tna_command()	489
14.172.1.3 send_tna_command()	489
14.173 /root/Desktop/Zappy/src/Server/help.c File Reference	489
14.173.1 Function Documentation	490
14.173.1.1 display_help()	490
14.174 /root/Desktop/Zappy/src/include/circular_buffer.h File Reference	491
14.174.1 Macro Definition Documentation	491
14.174.1.1 BUFFER_SIZE	491
14.174.2 Typedef Documentation	491
14.174.2.1 circular_buffer_t	491
14.174.3 Function Documentation	492
14.174.3.1 add_to_circular_buffer()	492
14.174.3.2 extract_command()	492
14.174.3.3 find_command_end()	492
14.174.3.4 init_circular_buffer()	492
14.175 /root/Desktop/Zappy/src/include/client.h File Reference	493
14.175.1 Typedef Documentation	493
14.175.1.1 client_t	494

14.175.1.2 graphical_client_t	494
14.175.2 Enumeration Type Documentation	494
14.175.2.1 client_type_e	494
14.175.3 Function Documentation	494
14.175.3.1 add_graphic_client()	494
14.175.3.2 remove_graphic_client()	494
14.175.3.3 send_map_info_to_one_client()	495
14.175.3.4 send_message_to_all_graphic()	495
14.176 /root/Desktop/Zappy/src/Server/include/command.h File Reference	495
14.176.1 Macro Definition Documentation	497
14.176.1.1 MAX_LIFE_AFTER FOOD	497
14.176.2 Typedef Documentation	497
14.176.2.1 command_data_t	497
14.176.3 Function Documentation	497
14.176.3.1 add_pending_cmd()	498
14.176.3.2 broadcast()	498
14.176.3.3 can_connect()	498
14.176.3.4 can_start_incantation()	499
14.176.3.5 check_player_starvation()	499
14.176.3.6 check_rota_tiles()	499
14.176.3.7 cleanup_client()	500
14.176.3.8 cleanup_pending()	500
14.176.3.9 cleanup_player_queue()	500
14.176.3.10 command_bct()	500
14.176.3.11 command_msz()	501
14.176.3.12 command_mtc()	501
14.176.3.13 command_pdi()	501
14.176.3.14 command_pdr()	501
14.176.3.15 command_pex()	502
14.176.3.16 command_pfk()	502
14.176.3.17 command_pgt()	503
14.176.3.18 command_pie()	503
14.176.3.19 command_pin()	503
14.176.3.20 command_plv()	504
14.176.3.21 command_ppo()	504
14.176.3.22 command_sst()	504
14.176.3.23 command_tna()	505
14.176.3.24 connect_nbr()	505
14.176.3.25 connect_nbr_srv()	505
14.176.3.26 determine_type()	505
14.176.3.27 eject()	506
14.176.3.28 execute_com()	506

14.176.3.29 find_client_by_id()	506
14.176.3.30 finish_incantation()	507
14.176.3.31 fork_c()	507
14.176.3.32 forward()	507
14.176.3.33 get_command_data()	508
14.176.3.34 get_message()	508
14.176.3.35 handle_player_death()	508
14.176.3.36 inventory()	509
14.176.3.37 left()	509
14.176.3.38 look()	509
14.176.3.39 process_next_queued_command()	510
14.176.3.40 right()	510
14.176.3.41 send_bct_to_all_graphical_clients()	510
14.176.3.42 send_enw_command()	511
14.176.3.43 send_enw_command_start()	511
14.176.3.44 send_mtc_to_all_graphical_clients()	511
14.176.3.45 send_pnw_command_to_all()	511
14.176.3.46 set_object()	512
14.176.3.47 start_incantation()	512
14.176.3.48 str_to_word_arr()	512
14.176.3.49 take_object()	513
14.176.3.50 tile_to_str()	513
14.176.3.51 write_command_output()	513
14.177 /root/Desktop/Zappy/src/Server/include/egg.h File Reference	514
14.177.1 Typedef Documentation	514
14.177.1.1 egg_t	514
14.177.1.2 server_t	514
14.177.1.3 tile_t	514
14.177.2 Function Documentation	514
14.177.2.1 add_egg()	515
14.177.2.2 create_egg()	515
14.177.2.3 remove_egg()	515
14.178 /root/Desktop/Zappy/src/Server/include/graphical_commands.h File Reference	515
14.178.1 Function Documentation	516
14.178.1.1 command_pbc()	516
14.178.1.2 command_pic()	517
14.178.1.3 command_seg()	517
14.178.1.4 command_sgt()	518
14.178.1.5 send_all_player_info_to_one_client()	518
14.178.1.6 send_ebo_command()	518
14.178.1.7 send_edi_command()	519
14.178.1.8 send_msz_command()	519

14.178.1.9 send_pin_command()	519
14.178.1.10 send_pin_to_all()	520
14.178.1.11 send_plv_command()	520
14.178.1.12 send_plv_to_all()	520
14.178.1.13 send_pnw_command()	520
14.178.1.14 send_ppo_command()	521
14.178.1.15 send_smg_command()	521
14.178.1.16 send_team_names_to_one_client()	521
14.178.1.17 send_tile_content_to_one_client()	522
14.178.1.18 send_tna_command()	522
14.179 /root/Desktop/Zappy/src/Server/include/parsing.h File Reference	522
14.179.1 Typedef Documentation	523
14.179.1.1 parsing_info_t	523
14.179.2 Function Documentation	523
14.179.2.1 arr_len()	523
14.179.2.2 free_arr()	523
14.179.2.3 parse_args()	524
14.179.2.4 parsing_error()	524
14.179.2.5 str_to_word_arr()	524
14.180 /root/Desktop/Zappy/src/Server/include/pending_cmd.h File Reference	524
14.180.1 Typedef Documentation	525
14.180.1.1 client_t	525
14.180.1.2 cmd_func_t	525
14.180.1.3 pending_cmd_t	525
14.180.1.4 server_t	525
14.181 /root/Desktop/Zappy/src/Server/include/pending_cmd_utils.h File Reference	525
14.181.1 Function Documentation	526
14.181.1.1 execute_pending_cmd()	526
14.182 /root/Desktop/Zappy/src/Server/include/player.h File Reference	527
14.182.1 Typedef Documentation	528
14.182.1.1 player_inventory_t	528
14.182.1.2 player_t	528
14.182.2 Enumeration Type Documentation	528
14.182.2.1 rotation_e	528
14.182.3 Function Documentation	528
14.182.3.1 add_item_to_inventory()	528
14.182.3.2 free_inventory()	529
14.182.3.3 get_inventory_content()	529
14.182.3.4 get_resource_name()	529
14.182.3.5 how_many_in_inventory()	529
14.182.3.6 init_inventory()	530
14.182.3.7 init_player()	530

14.182.3.8 inventory_has_item()	530
14.182.3.9 remove_item_from_inventory()	530
14.183 /root/Desktop/Zappy/src/Server/include/server.h File Reference	531
14.183.1 Typedef Documentation	532
14.183.1.1 poll_manager_t	532
14.183.1.2 server_t	532
14.183.2 Function Documentation	533
14.183.2.1 add_fd()	533
14.183.2.2 add_to_command_queue()	533
14.183.2.3 check_client()	533
14.183.2.4 count_team()	534
14.183.2.5 create_map()	534
14.183.2.6 create_server()	535
14.183.2.7 display_help()	535
14.183.2.8 finish_incantation()	535
14.183.2.9 free_all()	536
14.183.2.10 free_node()	536
14.183.2.11 init_new_player_pos()	537
14.183.2.12 init_server_eggs()	537
14.183.2.13 print_co()	537
14.183.2.14 process_next_queued_command()	538
14.183.2.15 remove_fd()	538
14.183.2.16 server_err()	539
14.183.2.17 update_game_tick()	539
14.184 /root/Desktop/Zappy/src/Server/include/tile.h File Reference	540
14.184.1 Macro Definition Documentation	540
14.184.1.1 DERAUMERE_D	540
14.184.1.2 FOOD_D	541
14.184.1.3 LINEMATE_D	541
14.184.1.4 MENDIANE_D	541
14.184.1.5 PHIRAS_D	541
14.184.1.6 SIBUR_D	541
14.184.1.7 THYSTAME_D	541
14.184.2 Typedef Documentation	541
14.184.2.1 resource_dist_t	541
14.184.2.2 resource_type_t	541
14.184.2.3 server_t	541
14.184.2.4 tile_t	541
14.184.3 Enumeration Type Documentation	542
14.184.3.1 resource_type	542
14.184.4 Function Documentation	542
14.184.4.1 add_egg_to_tile()	542

14.184.4.2 distribute_resources()	542
14.184.4.3 remove_egg_from_tile()	543
14.184.4.4 respawn_resources()	543
14.184.4.5 tile_add_player()	543
14.184.4.6 tile_free()	544
14.184.4.7 tile_init()	544
14.184.4.8 tile_remove_player()	544
14.185 /root/Desktop/Zappy/src/Server/main.c File Reference	544
14.185.1 Function Documentation	545
14.185.1.1 check_mini_args()	545
14.185.1.2 disp_args()	545
14.185.1.3 get_running_flag()	546
14.185.1.4 main()	546
14.185.1.5 server_loop()	546
14.185.1.6 set_rdm_seed()	547
14.185.1.7 setup_signal_handler()	547
14.185.1.8 should_continue_running()	547
14.185.1.9 sigint_handler()	547
14.185.1.10 sigpipe_handler()	548
14.186 /root/Desktop/Zappy/src/Server/map/egg.c File Reference	548
14.186.1 Function Documentation	548
14.186.1.1 add_egg()	548
14.186.1.2 add_egg_to_tile()	549
14.186.1.3 create_egg()	549
14.186.1.4 remove_egg()	549
14.186.1.5 remove_egg_by_id()	550
14.186.1.6 remove_egg_from_tile()	550
14.186.1.7 remove_id_from_array()	550
14.186.1.8 remove_node()	550
14.187 /root/Desktop/Zappy/src/Server/map/map_creation.c File Reference	551
14.187.1 Function Documentation	551
14.187.1.1 create_map()	551
14.188 /root/Desktop/Zappy/src/Server/map/resources_map.c File Reference	552
14.188.1 Function Documentation	552
14.188.1.1 distribute_one_resource()	553
14.188.1.2 distribute_resources()	553
14.188.1.3 init_resource_dist()	553
14.188.1.4 respawn_resources()	554
14.188.1.5 shuffle_indices()	554
14.189 /root/Desktop/Zappy/src/Server/map/tick_update.c File Reference	554
14.189.1 Function Documentation	555
14.189.1.1 tick_check()	555

14.189.1.2 update_game_tick()	555
14.190 /root/Desktop/Zappy/src/Server/map/tile.c File Reference	556
14.190.1 Function Documentation	556
14.190.1.1 tile_add_player()	556
14.190.1.2 tile_free()	557
14.190.1.3 tile_init()	557
14.190.1.4 tile_remove_player()	557
14.191 /root/Desktop/Zappy/src/Server/network/array_function.c File Reference	558
14.191.1 Function Documentation	558
14.191.1.1 add_token()	558
14.191.1.2 arr_len()	559
14.191.1.3 check_if_failed()	559
14.191.1.4 count_valid_tokens()	559
14.191.1.5 fill_result_array()	559
14.191.1.6 free_cpy_arr()	560
14.191.1.7 init_result_array()	560
14.191.1.8 next_valid_token()	560
14.191.1.9 process_tokens()	561
14.191.1.10 str_to_word_arr()	561
14.192 /root/Desktop/Zappy/src/Server/network/client_node.c File Reference	561
14.192.1 Function Documentation	562
14.192.1.1 cleanup_player_client()	562
14.192.1.2 free_node()	562
14.193 /root/Desktop/Zappy/src/Server/network/connection.c File Reference	563
14.193.1 Function Documentation	564
14.193.1.1 add_fd()	564
14.193.1.2 copy_names()	564
14.193.1.3 create_server()	565
14.193.1.4 init_new_client()	565
14.193.1.5 init_server()	565
14.193.1.6 init_server_resources()	566
14.193.1.7 init_server_socket()	566
14.193.1.8 remove_fd()	566
14.193.1.9 remove_head_client()	567
14.193.1.10 remove_other_client()	567
14.194 /root/Desktop/Zappy/src/Server/network/connection_err.c File Reference	567
14.194.1 Function Documentation	568
14.194.1.1 print_co()	568
14.194.1.2 server_err()	568
14.195 /root/Desktop/Zappy/src/Server/network/connection_utils.c File Reference	569
14.195.1 Function Documentation	569
14.195.1.1 count_team()	569

14.195.1.2 <code>create_server_egg()</code>	570
14.195.1.3 <code>init_server_eggs()</code>	570
14.196 /root/Desktop/Zappy/src/Server/network/graphical_client.c File Reference	570
14.196.1 Function Documentation	571
14.196.1.1 <code>add_graphic_client()</code>	571
14.196.1.2 <code>add_to_graphical_list()</code>	571
14.196.1.3 <code>remove_graphic_client()</code>	572
14.196.1.4 <code>remove_graphical_node()</code>	572
14.196.1.5 <code>send_map_info_to_one_client()</code>	572
14.196.1.6 <code>send_message_to_all_graphic()</code>	573
14.197 /root/Desktop/Zappy/src/Server/network/queue_connection.c File Reference	573
14.197.1 Function Documentation	573
14.197.1.1 <code>add_to_command_queue()</code>	573
14.197.1.2 <code>check_rebuild()</code>	574
14.197.1.3 <code>process_next_queued_command()</code>	574
14.198 /root/Desktop/Zappy/src/Server/network/server_run.c File Reference	575
14.198.1 Function Documentation	575
14.198.1.1 <code>check_client()</code>	575
14.198.1.2 <code>check_client_message()</code>	576
14.198.1.3 <code>check_new_connection()</code>	576
14.198.1.4 <code>fill_poll_array()</code>	576
14.198.1.5 <code>find_client_by_socket()</code>	576
14.198.1.6 <code>handle_game_tick()</code>	577
14.198.1.7 <code>new_connection()</code>	577
14.198.1.8 <code>poll_client()</code>	577
14.198.1.9 <code>setup_poll_manager()</code>	578
14.198.1.10 <code>smart_polling()</code>	578
14.199 /root/Desktop/Zappy/src/Server/parsing.c File Reference	578
14.199.1 Function Documentation	579
14.199.1.1 <code>check_args()</code>	579
14.199.1.2 <code>check_other_flags()</code>	580
14.199.1.3 <code>check_zero()</code>	580
14.199.1.4 <code>is_dup_flag()</code>	580
14.199.1.5 <code>parse_args()</code>	580
14.199.1.6 <code>parse_client_nb()</code>	581
14.199.1.7 <code>parse_coordinates()</code>	581
14.199.1.8 <code>parse_frequence()</code>	581
14.199.1.9 <code>parse_names()</code>	582
14.199.1.10 <code>parse_port()</code>	582
14.200 /root/Desktop/Zappy/src/Server/parsing_err.c File Reference	582
14.200.1 Function Documentation	583
14.200.1.1 <code>free_names()</code>	583

14.200.1.2 parsing_error()	583
14.201 /root/Desktop/Zappy/src/Server/player/player_death.c File Reference	583
14.201.1 Function Documentation	584
14.201.1.1 check_player_starvation()	584
14.201.1.2 handle_player_death()	584
14.202 /root/Desktop/Zappy/src/Server/player/player_init.c File Reference	585
14.202.1 Function Documentation	586
14.202.1.1 find_egg_for_team()	586
14.202.1.2 init_new_player_pos()	586
14.202.1.3 init_pending()	586
14.202.1.4 init_player()	586
14.202.1.5 init_player_team()	587
14.203 /root/Desktop/Zappy/src/Server/player/player_inventory.c File Reference	587
14.203.1 Function Documentation	588
14.203.1.1 add_item_to_inventory()	588
14.203.1.2 free_inventory()	588
14.203.1.3 init_inventory()	588
14.203.1.4 remove_item_from_inventory()	588
14.204 /root/Desktop/Zappy/src/Server/player/player_inventory_display.c File Reference	589
14.204.1 Function Documentation	589
14.204.1.1 add_separator()	589
14.204.1.2 append_item_string()	590
14.204.1.3 get_inventory_content()	590
14.204.1.4 get_one_item_content()	590
14.204.1.5 process_one_item()	590
14.205 /root/Desktop/Zappy/src/Server/player/player_inventory_utils.c File Reference	591
14.205.1 Function Documentation	591
14.205.1.1 get_resource_name()	591
14.205.1.2 how_many_in_inventory()	592
14.205.1.3 inventory_has_item()	592
14.206 /root/Desktop/Zappy/src/Shared/Common.hpp File Reference	592
14.207 /root/Desktop/Zappy/src/Shared/DLLoader.hpp File Reference	594
14.208 /root/Desktop/Zappy/src/Shared/IGraphicsLib.hpp File Reference	595
14.209 /root/Desktop/Zappy/src/Shared/IGuiLib.hpp File Reference	596
14.210 /root/Desktop/Zappy/src/Shared/LibraryManager.hpp File Reference	597
Index	599

Chapter 1

Zappy

Jeu multijoueur basé sur l'IA en réseau

Contrôlez des agents autonomes, collectez des ressources, évoluez et conquérez un monde régi par TCP et unités temporelles.

1.1 Table des matières

- [Vue d'ensemble](#)
 - [Architecture](#)
 - [Prérequis](#)
 - [Installation](#)
 - [Utilisation](#)
 - [Composants](#)
 - [Serveur](#)
 - [Interface Graphique](#)
 - [Intelligence Artificielle](#)
 - [Releases](#)
 - [Documentation](#)
 - [Développement](#)
-

1.2 Vue d'ensemble

[Zappy](#) est un projet multi-composants développé en C/C++ et Python, reproduisant un écosystème virtuel où des agents contrôlés par IA évoluent en collectant des ressources, en communiquant, et en réalisant des rituels d'élévation. Le jeu est composé de trois éléments principaux :

- **Serveur** : Écrit en C, il gère la logique du jeu, les règles, et l'état du monde
- **Interface Graphique** : Développée en C++, elle visualise l'état du jeu en temps réel
- **Clients IA** : Implémentés en Python, ils contrôlent automatiquement les joueurs pour accomplir des objectifs stratégiques

1.3 Architecture

Les composants communiquent via des protocoles réseau TCP/IP :

Commandes/Réponses

Client
(IA)

Serveur

Données de jeu

GUI

1.4 Prérequis

Pour compiler et exécuter [Zappy](#), vous aurez besoin de :

- Compilateur C/C++ (GCC ou Clang)
- Python 3.x
- Make
- Bibliothèques de développement graphique (pour le client GUI)

1.5 Installation

```
# Cloner le dépôt
git clone https://github.com/Epitech/Zappy.git
cd Zappy
# Compiler tous les composants
make
# Ou compiler des composants spécifiques
make zappy_server
make zappy_gui
make zappy_ia
```

1.6 Utilisation

1.6.1 Démarrer le serveur

```
./zappy_server -p <port> -x <width> -y <height> -n <team names> -c <max clients> -f <freq> [options]
```

Paramètres obligatoires :

- -p <port> : Port d'écoute du serveur
- -x <width> : Largeur de la carte (minimum 10)
- -y <height> : Hauteur de la carte (minimum 10)
- -n <team1> [team2] ... : Noms des équipes (séparés par des espaces)
- -c <max_clients> : Nombre maximum de clients par équipe
- -f <freq> : Fréquence du serveur (unités de temps par seconde)

Paramètres optionnels :

- --auto-start on|off : Démarrage automatique du jeu (défaut: off)
- --display-eggs true|false : Visibilité des ufs (défaut: true)
- --game_duration <time> : Durée de la partie en secondes
- -v ou --verbose : Mode verbeux pour le debug

Exemple :

```
./zappy_server -p 4242 -x 20 -y 20 -n team1 team2 team3 -c 5 -f 100 --auto-start on --display-eggs true
```

1.6.2 Démarrer l'interface graphique

```
./zappy_gui -p <port> -h <host>
```

Paramètres :

- **-p <port>** : Port du serveur
- **-h <host>** : Adresse du serveur (défaut: localhost)

Exemple :

```
./zappy_gui -p 4242 -h localhost
```

Contrôles de l'interface :

- **Caméra** : Clic gauche + glisser pour tourner, molette pour zoomer
- **Vue** : Touches WASD pour déplacer la caméra
- **Interface** : Panneaux d'information en temps réel sur les équipes et ressources

1.6.3 Démarrer l'IA

```
./zappy_ai -p <port> -n <team name> [-h <host>]
```

Paramètres :

- **-p <port>** : Port du serveur
- **-n <team>** : Nom de l'équipe
- **-h <host>** : Adresse du serveur (optionnel, défaut: localhost)

Exemple :

```
./zappy_ai -p 4242 -n team1
```

Comportements de l'IA :

- **Collecte de ressources** : Exploration automatique et collecte optimisée
- **Communication inter-agents** : Coordination via broadcast chiffré
- **Évolution** : Réalisation automatique des incantations pour monter de niveau
- **Stratégie adaptative** : Changement de comportement selon le contexte

1.7 Composants

1.7.1 Serveur

Le serveur [Zappy](#) est le cœur du système, développé en C pour des performances optimales.

Fonctionnalités principales :

- Gestion de la carte de jeu et des ressources
- Traitement des commandes clients en temps réel
- Simulation physique et temporelle du monde
- Protocoles de communication TCP/IP
- Gestion multi-threads pour la performance

Architecture modulaire :

- `network/` : Gestion des connexions et protocoles
- `map/` : Système de carte et tiles
- `player/` : Gestion des joueurs et leurs états
- `command/` : Traitement des commandes
- `buffer/` : Système de buffers pour la communication

1.7.2 Interface Graphique

Interface 3D développée en C++ utilisant Raylib pour la visualisation temps réel.

Fonctionnalités :

- Rendu 3D de la carte et des entités
- Interface utilisateur intuitive avec RayGUI
- Système de caméra libre avec contrôles fluides
- Affichage en temps réel des statistiques
- Gestion des textures et modèles 3D

Composants :

- renderer/ : Moteur de rendu 3D
- cameraController/ : Gestion de la caméra
- textureManager/ : Chargement et gestion des textures
- network/ : Communication avec le serveur
- graphicalContext/ : Contexte graphique et état

1.7.3 Intelligence Artificielle

Bots autonomes développés en Python avec une architecture modulaire et évolutive.

Caractéristiques :

- Architecture comportementale adaptative
- Système de communication sécurisé entre agents
- Algorithmes d'exploration et d'optimisation
- Stratégies d'évolution et de coopération
- Logging avancé pour le debug et l'analyse ./zappy_ai -p 4242 -n team1


```
## Composants
### Serveur
Le serveur est le composant central qui gère :
- La carte de jeu et les ressources
- Les connexions des clients (IA et GUI)
- La logique et les règles du jeu
- La gestion du temps et des événements
- Le protocole de communication
- Les rituels d'élévation et évolution des joueurs
### Interface Graphique
L'interface graphique offre :
- Visualisation en temps réel de la carte du jeu
- Affichage des joueurs et leurs inventaires
- Visualisation des ressources sur chaque case
- Suivi des événements (incantations, reproductions, etc.)
- Contrôles interactifs pour la navigation et observation
### Intelligence Artificielle
L'IA des clients comprend :
- Modules de perception pour l'analyse de l'environnement
- Algorithmes de décision stratégique
- Gestion des ressources et planification
- Communication et coordination entre agents
- Stratégies pour les rituels d'élévation
## Releases
Le projet utilise un système de release automatisé via GitHub Actions avec versioning sémantique :
### Télécharger une release
Rendez-vous sur la [page des releases](../../releases) pour télécharger la dernière version.
### Créer une nouvelle release
Les releases sont créées automatiquement via GitHub Actions :
1. **Accéder aux Actions** : GitHub onglet Actions workflow "Release Zappy"
2. **Choisir le type de version** :
   - 'nouvelle-version' : Version majeure (v1.0.0 v2.0.0)
   - 'nouvelle-feature' : Nouvelle fonctionnalité (v1.0.0 v1.1.0)
   - 'resolution-bug' : Correction de bug (v1.0.0 v1.0.1)
3. **Lancer le workflow** : Le système génère automatiquement la version, compile le projet et crée la release
```

```
Chaque release contient :
- Tous les binaires (serveur, GUI, IA)
- Bibliothèques statiques et dynamiques
- Script d'installation automatique
- Documentation et changelog
  **Guide détaillé** : [docs/release-workflow.md] (docs/release-workflow.md)
## Documentation
Une documentation détaillée est disponible dans le dossier 'docs/' :
- Architecture et conception : 'docs/architecture.md'
- Guide de démarrage : 'docs/getting-started.md'
- Documentation du serveur : 'docs/server/'
- Documentation de l'interface graphique : 'docs/gui/'
- Documentation de l'IA : 'docs/ai/'
La documentation Docusaurus peut être consultée localement :
```
cd docs/docusaurus
yarn start
```

```

1.8 Développement

1.8.1 Structure du projet

```
src/
  Server/      # Implémentation du serveur en C
  GUI/         # Interface graphique en C++
  AI/          # Client IA en Python
  Shared/       # Code partagé entre les composants
libs/           # Bibliothèques externes (RaylibCPP, RayGUICPP)
docs/          # Documentation du projet
tests/         # Tests unitaires et fonctionnels
assets/        # Ressources graphiques
tools/         # Outils de développement
```

1.8.2 Tests

```
# Exécuter tous les tests
make tests_run
# Tester le réseau GUI
make functional_tests
# Compiler en mode debug
make debug
```

Développé avec par l'équipe Zappy

© 2025 Epitech

Chapter 2

Utilisation de DLLoader et des bibliothèques dynamiques graphiques dans Zappy GUI

Ce dossier contient le code source de l'interface graphique de [Zappy](#). L'application utilise un système de chargement dynamique de bibliothèques ([DLLoader](#)) pour permettre l'utilisation de différentes implémentations graphiques et GUI sans modifier le code principal.

2.1 Fonctionnement général

- Les interfaces `IGraphicsLib` et `IGuiLib` définissent les fonctions attendues pour les bibliothèques graphiques et GUI.
- Le gestionnaire `LibraryManager` (pattern Singleton) permet de charger dynamiquement des bibliothèques partagées (.so) et d'accéder à leurs fonctionnalités via les interfaces.
- Les bibliothèques concrètes (ex : RaylibCPP, [RayGUICPP](#)) sont compilées séparément et placées dans le dossier `plugins/`.

2.2 Exemple d'utilisation

```
#include "Shared/LibraryManager.hpp"
// Chargement des bibliothèques dynamiques
auto& libraryManager = LibraryManager::getInstance();
libraryManager.loadGraphicsLib("plugins/libraylibcpp.so");
libraryManager.loadGuiLib("plugins/libraygui.so");
// Récupération des interfaces
auto& graphics = libraryManager.getGraphicsLib();
auto& gui = libraryManager.getGuiLib();
// Utilisation
graphics.InitWindow(1280, 720, "Zappy");
gui.DrawButton(100, 100, 200, 50, "Lancer la partie");
```

2.3 Ajouter une nouvelle bibliothèque

1. Implémentez l'interface `IGraphicsLib` ou `IGuiLib` dans une nouvelle classe.
2. Exposez une fonction de création compatible C :

```
extern "C" {
    IGraphicsLib* createGraphicsLib() {
        static MaLib instance;
        return &instance;
    }
}
```
3. Compilez la bibliothèque en .so et placez-la dans `plugins/`.
4. Chargez-la via le `LibraryManager` dans votre code.

2.4 Avantages

- Permet de changer de moteur graphique ou GUI sans recompiler toute l'application.
- Favorise l'extensibilité et la modularité du projet.

2.5 Exemple minimal

Voir `test_dlloader.cpp` pour un exemple de code qui charge dynamiquement les bibliothèques graphiques et GUI, puis affiche une fenêtre avec un bouton.

2.5.1 Compilation

Assurez-vous d'avoir compilé les bibliothèques dynamiques :

```
make -C ../../libs/RaylibCPP  
make -C ../../libs/RayGUICPP
```

Puis compilez l'exemple :

```
g++ -I../../libs/RaylibCPP -I../../libs/RayGUICPP/include -I../Shared -o test_dlloader test_dlloader.cpp  
-ldl -lraylib -lGL -lm -lpthread -lrt -lX11
```

2.5.2 Exécution

Placez les fichiers `libraylibcpp.so` et `libraygui.so` dans le dossier `plugins/` à la racine du projet, puis lancez :

```
./test_dlloader
```

2.6 Fonctionnement

- Le `LibraryManager` charge dynamiquement les bibliothèques graphiques et GUI via leurs interfaces (`IGraphicsLib`, `IGuiLib`).
- Vous pouvez ensuite utiliser les méthodes de ces interfaces comme si elles étaient natives.

2.7 Pour aller plus loin

- Consultez le README dans `src/Shared/` pour plus de détails sur le système de plugins.
- Adaptez l'exemple pour tester vos propres bibliothèques dynamiques !

Chapter 3

Serveur Zappy

3.1 Vue d'ensemble

Le serveur [Zappy](#) est le composant central du projet, développé en C pour des performances optimales. Il gère l'état du jeu, les connexions clients multiples, et applique toutes les règles de simulation du monde virtuel.

3.2 Architecture

3.2.1 Composants Principaux

- **Gestionnaire de réseau** (`network/`) - Communication TCP/IP avec clients IA et GUI
- **Moteur de jeu** (`map/, player/`) - État du monde et règles de simulation
- **Processeur de commandes** (`command/`) - Interprétation et exécution des actions
- **Gestionnaire temporel** - Synchronisation et gestion des ticks de jeu

3.2.2 Structures de Données Clés

```
typedef struct server_s {
    int s_fd;                                // Socket serveur
    client_t *clients;                        // Clients IA connectés
    graphical_client_t *gui_clients;          // Clients GUI connectés
    tile_t **map;                             // Carte du monde
    int current_tick;                         // Tick actuel
    parsing_info_t *config;                   // Configuration
} server_t;
```

3.3 Compilation

```
cd src/Server
make           # Compilation standard
make debug     # Compilation avec symboles de debug
make clean      # Nettoyage des fichiers objets
```

3.4 Utilisation

3.4.1 Démarrage Standard

```
./zappy_server -p 4242 -x 20 -y 20 -n team1 team2 -c 5 -f 100
```

3.4.2 Paramètres

- `-p <port>` : Port d'écoute (requis)
- `-x <width>` : Largeur de la carte (requis, min: 10)
- `-y <height>` : Hauteur de la carte (requis, min: 10)

- **-n <teams>** : Noms des équipes séparés par des espaces (requis)
- **-c <clients>** : Nombre max de clients par équipe (requis)
- **-f <freq>** : Fréquence du serveur en ticks/seconde (requis)
- **--auto-start on|off** : Démarrage automatique (optionnel)
- **--display-eggs true|false** : Visibilité des ufs (optionnel)
- **--game_duration <sec>** : Durée de partie en secondes (optionnel)
- **-v, --verbose** : Mode verbeux (optionnel)

3.4.3 Exemples Avancés

```
# Partie rapide de test
./zappy_server -p 4242 -x 10 -y 10 -n test -c 2 -f 1000 --auto-start on
# Partie équilibrée multi-équipes
./zappy_server -p 4242 -x 30 -y 30 -n alpha beta gamma -c 6 -f 100 \
--game_duration 3600 --display-eggs true -v
# Configuration pour tournoi
./zappy_server -p 4242 -x 50 -y 50 -n team1 team2 team3 team4 -c 10 -f 100 \
--auto-start off --game_duration 7200
```

3.5 Protocoles de Communication

3.5.1 Clients IA

Le serveur accepte les connexions IA qui s'authentifient avec leur nom d'équipe et reçoivent des commandes de jeu. Voir [Protocole Serveur-IA](#) pour les détails.

3.5.2 Clients GUI

Les clients graphiques s'identifient avec **GRAPHIC** et reçoivent tous les événements de jeu en temps réel. Voir [Protocole Serveur-GUI](#).

3.6 Performance et Optimisation

3.6.1 Métriques

Le serveur peut gérer :

- **100+ clients simultanés** sur hardware moderne
- **1000+ ticks/seconde** pour des parties rapides
- **Latence < 1ms** pour les commandes locales
- **Mémoire < 100MB** pour des cartes de taille standard

3.6.2 Optimisations Implémentées

- **Polling non-bloquant** avec `poll()` pour les E/O réseau
- **Files de commandes** pour traitement asynchrone
- **Gestion mémoire optimisée** avec pools d'objets
- **Cache de map** pour éviter les recalculs

3.7 Tests

3.7.1 Tests Unitaires

```
make tests_run      # Tous les tests
make test_player    # Tests du module joueur
make test_map        # Tests du module carte
make test_commands   # Tests des commandes
```

3.7.2 Tests de Performance

```
# Test de charge avec clients simulés
./tests/stress_test.sh 100 clients
# Test de mémoire
valgrind --leak-check=full ./zappy_server -p 4242 -x 10 -y 10 -n test -c 1 -f 100
# Profiling CPU
perf record ./zappy_server -p 4242 -x 20 -y 20 -n team1 team2 -c 5 -f 100
perf report
```

3.8 Debugging

3.8.1 Mode Debug

```
# Compilation avec symboles
make debug
# Debugging avec GDB
gdb ./zappy_server
(gdb) run -p 4242 -x 10 -y 10 -n team1 -c 1 -f 100
(gdb) break main
(gdb) continue
```

3.8.2 Logs Verbeux

```
# Activation du mode verbeux
./zappy_server -v -p 4242 -x 10 -y 10 -n team1 -c 1 -f 100
# Redirection vers fichier
./zappy_server -v -p 4242 -x 10 -y 10 -n team1 -c 1 -f 100 > server.log 2>&1
```

3.9 Architecture Interne

3.9.1 Boucle Principale

```
while (server->should_run) {
    // 1. Poling des sockets clients
    poll(server->poll_manager->fds, server->nfds, timeout);

    // 2. Traitement des nouvelles connexions
    check_client(server);

    // 3. Lecture des commandes clients
    process_client_messages(server);

    // 4. Mise à jour du tick de jeu
    update_game_tick(server);

    // 5. Traitement des commandes en attente
    process_command_queues(server);

    // 6. Envoi des mises à jour aux GUI
    send_gui_updates(server);
}
```

3.9.2 Gestion des Commandes

```
typedef struct command_s {
    char *name;           // Nom de la commande
    int (*execute)(server_t *, client_t *, char **); // Fonction d'exécution
    int time_cost;         // Coût en ticks
    bool gui_only;        // Commande réservée aux GUI
} command_t;
```

3.9.3 États des Joueurs

Les joueurs progressent à travers différents états :

1. **Connecté** - Authentifié mais pas encore spawné
2. **Actif** - En jeu, peut exécuter des commandes
3. **Incantation** - En cours d'élévation (immobile)
4. **Mort** - Plus de nourriture, connexion fermée

3.10 Configuration Avancée

3.10.1 Variables d'Environnement

```
export ZAPPY_MAX_CLIENTS=200      # Limite de clients globale
export ZAPPY_TICK_PRECISION=1000   # Précision des ticks (ts)
export ZAPPY_LOG_LEVEL=INFO       # Niveau de log (DEBUG/INFO/WARN/ERROR)
export ZAPPY_BUFFER_SIZE=8192     # Taille des buffers réseau
```

3.10.2 Fichier de Configuration

Le serveur peut lire un fichier de configuration optionnel :

```
# zappy.conf
[server]
max_clients = 200
tick_precision = 1000
log_level = INFO
[game]
default_food_generation = 50
default_resource_density = 0.15
incantation_duration = 300
[network]
buffer_size = 8192
timeout = 5000
keep_alive = true
```

3.11 Monitoring et Maintenance

3.11.1 Signaux Système

- **SIGINT/SIGTERM** : Arrêt propre du serveur
- **SIGUSR1** : Rechargement de la configuration
- **SIGUSR2** : Dump de l'état interne

3.11.2 Métriques Runtime

```
# Envoi de signal pour statistiques
kill -USR2 <pid_serveur>
# Sortie dans les logs :
# [INFO] Runtime Statistics:
# - Connected clients: 25
# - Commands/sec: 150
# - Memory usage: 45MB
# - Uptime: 02:34:17
```

3.12 Contribuer

3.12.1 Standards de Code

- **Format** : K&R style, indentation 4 espaces
- **Nommage** : snake_case pour fonctions et variables
- **Documentation** : Commentaires Doxygen obligatoires
- **Tests** : Couverture > 80% pour nouvelles fonctionnalités

3.12.2 Workflow

1. Fork du dépôt
2. Branche feature : `git checkout -b feature/nouvelle-fonctionnalite`
3. Développement avec tests
4. Pull Request vers `develop`

3.13 Documentation Complète

- **Architecture Serveur** - Architecture détaillée
- **Protocole IA** - Communication avec les IA
- **API Doxygen** - Documentation générée du code

Chapter 4

RayGUICPP - Encapsulation C++ de raygui pour Zappy

Ce dossier fournit une encapsulation moderne de la bibliothèque raygui (UI pour Raylib) en C++.

4.1 Prérequis

- Raylib installé sur votre système
- Compilation sous Linux (X11, pthread, dl, rt, GL)

4.2 Structure

- `include/RayGuiEncap.hpp` : encapsulation C++ de raygui
- `raygui/src/raygui.h` : header raygui (fourni)
- `RayGui.cpp` et `raygui_impl.cpp` : implémentation C++ et pont avec raygui
- `test_rayguicpp.cpp` : exemple d'utilisation

4.3 Compilation

Placez-vous dans `libs/RayGUICPP` puis compilez avec :

```
g++ -Iinclude -Iraygui/src -o test_rayguicpp test_rayguicpp.cpp raygui_impl.cpp RayGui.cpp -lraylib -lGL -lm -lpthread -ldl -lrt -lX11
```

4.4 Exemple minimal

```
#include <raylib.h>
#include "RayGuiEncap.hpp"
int main() {
    InitWindow(800, 600, "Test RayGUICPP");
    while (!WindowShouldClose()) {
        BeginDrawing();
        ClearBackground(RAYWHITE);
        RayGUICPP::Button::Draw({100,100,200,40}, "Hello Raygui");
        EndDrawing();
    }
    CloseWindow();
    return 0;
}
```

4.5 Utilisation dans Zappy

- Inclure `RayGuiEncap.hpp` dans vos fichiers C++

- Utiliser les classes statiques du namespace `RayGUICPP` pour dessiner vos contrôles GUI (Button, Label, CheckBox, Slider, etc.)
- Voir `test_rayguicpp.cpp` pour un exemple complet

4.6 Intégration dynamique (DLLoader)

- `RayGUICPP` implémente l'interface `IGuiLib` (voir `src/Shared/IGuiLib.hpp`).
- Exporte la fonction suivante pour l'intégration dynamique :

```
extern "C" {
    std::shared_ptr<IGuiLib> createGuiLib();
}
```

- À compiler en bibliothèque partagée (`libraygui.so`) et placer dans le dossier `plugins/`.
- À charger via le `LibraryManager` de `Zappy` (voir README dans `src/Shared/`).

4.7 Exemple d'utilisation avec DLLoader

Voir `src/GUI/test_dlloader.cpp` pour un exemple d'utilisation dynamique.

4.8 Astuces

- Pour ajouter de nouveaux contrôles, suivez le modèle des classes existantes dans `RayGuiEncap.hpp`
- Pour personnaliser le style, utilisez `RayGui::LoadStyle("chemin/vers/style.rgs")`
- Pour compiler dans un autre dossier, adaptez les chemins d'inclusion `-I`

4.9 Conseils d'extension

- Pour ajouter un nouveau contrôle, créez une nouvelle classe dans `RayGuiEncap.hpp` et exposez-la dans le namespace `RayGUICPP`.
- Respectez l'interface pour garantir la compatibilité avec le système de plugins.

4.10 Liens utiles

- Documentation `raygui`
- Documentation `raylib`

Pour toute question, contactez l'équipe `Zappy` !

Chapter 5

RaylibCPP - Encapsulation C++ de Raylib

Ce dossier propose une encapsulation complète de Raylib en C++ pour le projet [Zappy](#).

5.1 Fonctionnalités principales

- Gestion de la fenêtre (création, fermeture, drawing)
- Dessin de formes 2D et 3D (rectangle, cercle, cube, sphère, grille, cylindre, ligne 3D, etc.)
- Gestion avancée des caméras 2D et 3D (création, modes, manipulation)
- Gestion des textures 2D et 3D (chargement, binding, déchargement)
- Gestion des modèles 3D (chargement, dessin, déchargement)
- Gestion des polices et rendu de texte
- Gestion de l'audio (sons, musiques, volume)
- Gestion des entrées clavier/souris

5.2 Structure du dossier

- window / : gestion de la fenêtre
- shape / : formes 2D/3D
- texture / : textures 2D/3D et modèles 3D
- font / : polices et texte
- audio / : sons et musiques
- input / : entrées clavier/souris
- camera / : caméras 2D/3D
- utils / : utilitaires (vecteurs, couleurs, etc.)

5.3 Exemple d'utilisation

```
#include "RayLib.hpp"
int main() {
    RayLib gfx;
    gfx.InitWindow(800, 600, "Demo RaylibCPP");
    gfx.CreateCamera3D({0,10,10}, {0,0,0}, {0,1,0}, 45.0f, 0);
    while (!gfx.WindowShouldClose()) {
        gfx.BeginDrawing();
        gfx.ClearBackground({255,255,255,255});
        gfx.BeginCamera3D();
        gfx.DrawCube({0,0,0}, 2,2,2, {255,0,0,255});
```

```
    gfx.EndCamera3D();
    gfx.EndDrawing();
}
gfx.CloseWindow();
return 0;
}
```

5.4 Intégration dynamique (DLLoader)

Cette bibliothèque est conçue pour être chargée dynamiquement via le système [DLLoader](#) de [Zappy](#).

- Implémente l'interface [IGraphicsLib](#) (voir `src/Shared/IGraphicsLib.hpp`).
- Exporte la fonction suivante pour l'intégration dynamique :

```
extern "C" {
    std::shared_ptr<IGraphicsLib> createGraphicsLib();
}
```

- À compiler en bibliothèque partagée (`libraylibcpp.so`) et placer dans le dossier `plugins/`.
- À charger via le [LibraryManager](#) de [Zappy](#) (voir `README` dans `src/Shared/`).

5.5 Exemple d'utilisation avec DLLoader

Voir `src/GUI/test_dlloader.cpp` pour un exemple d'utilisation dynamique.

5.6 Notes avancées

- Toute nouvelle fonctionnalité Raylib doit être encapsulée dans une classe dédiée (voir structure du dossier).
- Pour étendre, suivez le modèle des classes existantes et exposez les méthodes via l'interface.

Pour toute question, contactez l'équipe [Zappy](#) !

Chapter 6

Protocole de Communication Serveur-IA

6.1 Vue d'ensemble

Ce document détaille le protocole de communication entre le serveur [Zappy](#) et les clients d'intelligence artificielle (IA). Le protocole est basé sur des échanges textuels avec des commandes spécifiques et leurs réponses associées.

6.2 Connexion

1. L'IA se connecte au serveur via TCP/IP
2. Le serveur répond avec : WELCOME\n
3. L'IA envoie le nom de son équipe : <team_name>\n
4. Le serveur répond avec :
 - <client_num>\n : Numéro du client dans l'équipe
 - <X> <Y>\n : Dimensions de la carte

6.3 Format des commandes

- Chaque commande est une chaîne de caractères terminée par \n
- Les réponses du serveur sont également terminées par \n
- Chaque commande a un temps d'exécution spécifique

6.4 Commandes disponibles

6.4.1 Commandes de base

Commande	Description	Résultat	Durée (unité temps serveur)
Forward	Avancer d'une case	ok\n	7
Right	Pivoter à droite	ok\n	7
Left	Pivoter à gauche	ok\n	7
Look	Observer l'environnement	Liste des objets visibles	7
Inventory	Consulter l'inventaire	Liste des ressources possédées	1
Broadcast <text>	Communiquer avec les autres joueurs	ok\n	7

Commande	Description	Résultat	Durée (unité temps serveur)
Connect_nbr	Nombre de connexions disponibles	Nombre entier	0
Fork	Créer un nouvel uf	ok\n	42
Eject	Éjecter les joueurs de la case	ok\n ou ko\n	7
Take <object>	Ramasser un objet	ok\n ou ko\n	7
Set <object>	Poser un objet	ok\n ou ko\n	7
Incantation	Lancer une incantation	Succès ou échec de l'incantation	Variable

6.4.2 Format des réponses

6.4.2.1 Look

```
<object1> <object2> ...],  
[<object1> <object2> ...],  
...
```

Objets possibles : player, food, linemate, deraumere, sibur, mendiane, phiras, thystame, egg

6.4.2.2 Inventory

```
{food <n>, linemate <n>, deraumere <n>, sibur <n>, mendiane <n>, phiras <n>, thystame <n>}
```

6.4.2.3 Broadcast

- Les joueurs reçoivent : message <direction>, <message>\n
- Direction : nombre de 1 à 8 représentant la direction du message

6.4.2.4 Incantation

- Serveur répond immédiatement : Elevation underway\n
- À la fin de l'incantation : Current level: <level>\n ou échec

6.5 Conditions d'élévation

Niveau	Joueurs requis	Linemate	Deraumere	Sibur	Mendiane	Phiras	Thystame
12	1	1	0	0	0	0	0
23	2	1	1	1	0	0	0
34	2	2	0	1	0	2	0
45	4	1	1	2	0	1	0
56	4	1	2	1	3	0	0
67	6	1	2	3	0	1	0
78	6	2	2	2	2	2	1

6.6 Mort du joueur

Si un joueur meurt (manque de nourriture), le serveur ferme la connexion.

6.7 Exemple d'échange

```
CLIENT: <connexion>
SERVER: WELCOME\n
CLIENT: team1\n
SERVER: 0\n
```

```
SERVER: 8 8\nCLIENT: Look\nSERVER: [player food],[],[food],\nCLIENT: Forward\nSERVER: ok\nCLIENT: Inventory\nSERVER: {food 9, linemate 0, deraumere 0, sibur 0, mendiane 0, phiras 0, thystame 0}\n
```


Chapter 7

Documentation du Projet Zappy

7.1 Introduction

Bienvenue dans la documentation du projet [Zappy](#). Ce projet est un écosystème virtuel multi-agents composé de plusieurs composants interagissant ensemble pour créer un environnement de simulation complexe et performant. [Zappy](#) reproduit un monde virtuel où des agents contrôlés par intelligence artificielle évoluent, collectent des ressources, communiquent entre eux et tentent d'atteindre le niveau maximal par le biais d'incantations collaboratives.

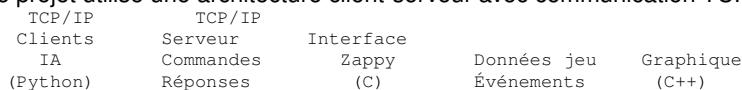
7.2 Vue d'ensemble du Système

7.2.1 Composants Principaux

- **Serveur** - Cur du système gérant la logique du jeu, écrit en C pour des performances optimales
- **Interface Graphique** - Visualisation 3D en temps réel développée en C++ avec Raylib
- **Intelligence Artificielle** - Agents autonomes développés en Python avec comportements adaptatifs
- **Bibliothèques Partagées** - Système de chargement dynamique et interfaces communes

7.2.2 Architecture Technique

Le projet utilise une architecture client-serveur avec communication TCP/IP :



7.3 Structure de la Documentation

7.3.1 Guides d'Utilisation

- **Guide de démarrage** - Installation, compilation et premiers pas
- **Architecture** - Vue d'ensemble technique du système
- **Guide de développement** - Standards de code, tests et contribution

7.3.2 Documentation Technique

7.3.2.1 Composants Principaux

- **Documentation Serveur** - Architecture, API et implémentation du serveur
- **Documentation IA** - Agents intelligents, comportements et stratégies

- **Documentation GUI** - Interface graphique 3D et système de rendu
- **Documentation Shared** - Système de chargement dynamique de bibliothèques

7.3.2.2 Bibliothèques et Outils

- **RaylibCPP** - Encapsulation C++ moderne de Raylib
- **RayGUICPP** - Interface utilisateur avec RayGUI
- **Historique raygui** - Évolution de la bibliothèque raygui

7.3.3 Protocoles de Communication

- **Protocole Serveur-IA** - Communication entre serveur et agents IA
- **Protocole Serveur-GUI** - Communication pour l'affichage graphique

7.3.4 Documentation Avancée

- **Patterns de conception** - Patterns utilisés dans le moteur de rendu
- **Implémentation du rendu** - Détails techniques du système de rendu

7.4 Documentation Interactive

7.4.1 Docusaurus (Documentation Web)

- **Localisation** : /docs/docusaurus/
- **Démarrage** : cd docs/docusaurus && npm start
- **URL locale** : `http://localhost:3000`

La documentation Docusaurus offre une expérience de navigation moderne avec :

- Interface responsive et recherche intégrée
- Support multi-langues (français/anglais)
- Navigation structurée par composants
- Exemples de code avec coloration syntaxique

7.4.2 Doxygen (Documentation API)

- **Configuration** : /docs/Doxygen/Doxyfile
- **Génération** : cd docs/Doxygen && doxygen
- **Sortie** : /docs/Doxygen/build/html/

La documentation Doxygen génère automatiquement :

- Documentation complète des APIs C/C++
- Diagrammes de classes et de dépendances
- Index des fonctions et structures
- Documentation extraite des commentaires du code

7.5 Démarrage Rapide

7.5.1 Installation des Dépendances

```
# Système (Ubuntu/Debian)
sudo apt update
sudo apt install build-essential cmake python3-dev python3-pip
sudo apt install libgl1-mesa-dev libglu1-mesa-dev libxrandr-dev
# Documentation
sudo apt install doxygen graphviz nodejs npm
# Python (IA)
cd src/AI && pip install -r requirements.txt
# Docusaurus
cd docs/docusaurus && npm install
```

7.5.2 Compilation et Test

```
# Compilation complète
make
# Test des composants
./zappy_server --help
./zappy_gui --help
python3 src/AI/main.py --help
# Tests unitaires
make tests_run
# Génération de la documentation
cd docs/Doxxygen && doxygen
cd docs/docusaurus && npm run build
```

7.6 Contribution à la Documentation

7.6.1 Standards de Documentation

1. **Markdown** : Format standard pour tous les documents
2. **Commentaires Doxygen** : Dans le code C/C++ pour génération automatique
3. **Docstrings Python** : Format Google/Numpy pour les modules Python
4. **Exemples** : Code fonctionnel dans tous les guides

7.6.2 Workflow de Contribution

1. **Branching** : Créer une branche `docs/feature-name`
2. **Édition** : Modifier/ajouter la documentation
3. **Vérification** : Tester la génération (Doxygen, Docusaurus)
4. **Pull Request** : Soumettre pour review

7.6.3 Organisation des Fichiers

```
docs/
index.md          # Cette page d'accueil
protocol_server_ai.md # Protocole de communication IA
design_patterns_*.md # Patterns de conception
docusaurus/
  docs/           # Documentation web interactive
  src/            # Composants et thèmes
  sidebars.ts     # Navigation
  Doxygen/        # Configuration et sortie API
    Doxyfile      # Configuration Doxygen
  build/          # Documentation générée
  assets/         # Images et ressources
```

7.7 Ressources Complémentaires

7.7.1 Références Externes

- **Raylib** - Bibliothèque graphique utilisée

- **Docusaurus** - Générateur de documentation
- **Doxygen** - Générateur de documentation API

7.7.2 Support et Communauté

- **Issues** : Signalement de bugs et demandes de fonctionnalités
- **Discussions** : Questions générales et aide communautaire
- **Wiki** : Documentation collaborative et tutoriels avancés

7.7.3 Licence et Copyright

Ce projet est développé par l'équipe Epitech dans le cadre du projet pédagogique B-YEP-400. Consultez le fichier LICENSE pour les détails de licence.

Chapter 8

Namespace Index

8.1 Namespace List

Here is a list of all namespaces with brief descriptions:

actions	41
agent	43
behaviors	43
broadcastManager	43
decisionManager	43
encryption	43
input	44
logger	45
main	45
paringArgsClass	46
RayGUICPP	46
raylibcpp	47
raylibcpp::Text3DHelper	48
resources	51
roles	52
socketManager	52
SystemWrapper	52
TypeAdapter	56
upgrades	58
zappy	58
Zappy	60
ZappyTypes	60
ZappyTypes::Colors	62

Chapter 9

Hierarchical Index

9.1 Class Hierarchy

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

agent.Agent	69
raylibcpp::Audio	74
broadcastManager.BroadcastManager	78
RayGUICPP::Button	79
raylibcpp::Camera2DWrap	80
raylibcpp::Camera3DWrap	81
CameraController	82
RayGUICPP::CheckBox	87
circular_buffer_s	87
client_s	88
ZappyTypes::Color	89
RayGUICPP::ColorPicker	90
RayGUICPP::ComboBox	91
command_data_s	91
decisionManager.DecisionManager	92
DLLoader< T >	95
RayGUICPP::DropdownBox	97
egg_s	99
Exception	
paringArgsClass.ParsingArgsError	193
std::exception	
IException	120
AEException	67
raylibcpp::Font	105
raylibcpp::GlyphDimensions	113
graphical_client_s	114
GraphicalContext	114
RayGUICPP::GroupBox	119
RayGUICPP::Icon	120
IGraphicalContextObserver	122
GameLoop	106
Zappy::MapRendererObserver	160
IGraphicsLib	123
RayLib	228
IGuiLib	132
RayGui	218
IMessageData	139
BroadcastData	76
EggData	101

EndGameData	104
IncantationData	141
IncantationEndData	143
MapSizeData	161
PlayerInfoData	200
PlayerInventoryData	203
ResourceData	248
ServerMessageData	259
TeamNameData	273
TileContentData	294
TimeUnitData	300
raylibcpp::Input	146
Zappy::ITileRenderStrategy	147
Zappy::DetailedTileRenderStrategy	93
Zappy::ModelTileRenderStrategy	177
Zappy::SimpleTileRenderStrategy	262
RayGUICPP::Label	148
LibraryManager	149
RayGUICPP::ListView	152
logger.Logger	153
Zappy::MapRenderer	156
Message	163
RayGUICPP::MessageBox	166
MessageQueue	167
Model3D	169
ModelManager	170
Zappy::ModelManagerAdapter	176
raylibcpp::ModelWrap	179
raylibcpp::MusicWrap	180
NetworkLogger	182
NetworkManager	183
NetworkThread	186
RayGUICPP::Panel	187
paringArgsClass.parseArgs	188
parsing_info_s	192
ParsingCLI	194
pending_cmd_s	196
player_inventory_s	197
player_s	197
poll_manager_s	207
RayGUICPP::ProgressBar	208
ProtocolParser	209
RayGUICPP::RayGui	227
ZappyTypes::Rectangle	242
Renderer	243
raylibcpp::RenderState	246
resource_dist_s	247
SystemWrapper::SafeBuffer	250
SystemWrapper::SafePollFd	251
SystemWrapper::SafeSockAddr	253
RayGUICPP::ScrollPane	255
server_s	256
raylibcpp::Shape	260
RayGUICPP::Slider	264
socketManager.SocketManager	264
raylibcpp::SoundWrap	267
RayGUICPP::Spinner	269
RayGUICPP::StatusBar	269

RayGUICPP::TabBar	270
TcpConnection	270
raylibcpp::Text3D	274
raylibcpp::Text3DCodepoint	276
raylibcpp::Text3DConfig	277
raylibcpp::Text3DMeasurement	280
raylibcpp::Text3DParser	281
raylibcpp::Text3DRenderer	282
raylibcpp::Text3DWave	283
RayGUICPP::TextBox	284
RayGUICPP::TextInputBox	285
raylibcpp::Texture	286
raylibcpp::Texture3DWrap	287
raylibcpp::TextureCoordinates	288
TextureManager	289
tile_s	293
TileData	298
Zappy::TileRenderStrategyFactory	299
RayGUICPP::Toggle	302
RayGUICPP::ToggleGroup	302
UIRenderer	303
raylibcpp::Utils	306
RayGUICPP::ValueBox	307
ZappyTypes::Vector2	307
ZappyTypes::Vector3	308
raylibcpp::Vector3Wrap	309
raylibcpp::WaveTextConfig	311
raylibcpp::Window	311
RayGUICPP::WindowBox	313
ABC	
behaviors.Behavior	75
behaviors.DysonBehavior	98
behaviors.GetFoodAndMineralsBehavior	109
behaviors.GetFoodBehavior	111
behaviors.GetMineralsBehavior	112
behaviors.UpgradeBehavior	305
IntEnum	
logger.LogLevel	155

Chapter 10

Data Structure Index

10.1 Data Structures

Here are the data structures with brief descriptions:

AException	67
agent.Agent	69
raylibcpp::Audio	74
behaviors.Behavior	75
BroadcastData	76
broadcastManager.BroadcastManager	78
RayGUICPP::Button	79
raylibcpp::Camera2DWrap	80
raylibcpp::Camera3DWrap	81
CameraController	82
RayGUICPP::CheckBox	87
circular_buffer_s	87
client_s	88
ZappyTypes::Color	89
RayGUICPP::ColorPicker	90
RayGUICPP::ComboBox	91
command_data_s	91
decisionManager.DecisionManager	92
Zappy::DetailedTileRenderStrategy	93
DLLoader< T >	95
RayGUICPP::DropdownBox	97
behaviors.DysonBehavior	98
egg_s	99
EggData	101
EndGameData	104
raylibcpp::Font	105
GameLoop	106
behaviors.GetFoodAndMineralsBehavior	109
behaviors.GetFoodBehavior	111
behaviors.GetMineralsBehavior	112
raylibcpp::GlyphDimensions	113
graphical_client_s	114
GraphicalContext	114
RayGUICPP::GroupBox	119
RayGUICPP::Icon	120
IException	120
IGraphicalContextObserver	
Observateur de changements de contexte graphique	122
IGraphicsLib	123
IGuiLib	132

IMessageData	139
IncantationData	141
IncantationEndData	143
raylibcpp::Input	146
Zappy::ITileRenderStrategy	147
RayGUICPP::Label	148
LibraryManager	149
RayGUICPP::ListView	152
logger.Logger	153
logger.LogLevel	155
Zappy::MapRenderer	
Renderer de carte responsable de l'affichage de la grille Implémente le pattern Bridge pour séparer l'interface du rendu	156
Zappy::MapRendererObserver	160
MapSizeData	161
Message	163
RayGUICPP::MessageBox	166
MessageQueue	167
Model3D	
Structure représentant un modèle 3D avec ses textures associées	169
ModelManager	
Gestionnaire de modèles 3D singleton qui implémente un pattern de pool de ressources pour éviter de charger plusieurs fois les mêmes modèles	170
Zappy::ModelManagerAdapter	
Adaptateur qui permet de gérer ModelManager comme un std::shared_ptr Cette classe encapsule le singleton ModelManager pour l'utiliser avec des shared_ptr	176
Zappy::ModelTileRenderStrategy	177
raylibcpp::ModelWrap	179
raylibcpp::MusicWrap	180
NetworkLogger	182
NetworkManager	183
NetworkThread	186
RayGUICPP::Panel	187
paringArgsClass.parseArgs	188
parsing_info_s	192
paringArgsClass.ParsingArgsError	193
ParsingCLI	194
pending_cmd_s	196
player_inventory_s	197
player_s	197
PlayerInfoData	200
PlayerInventoryData	203
poll_manager_s	
Structure de gestion du polling non-bloquant	207
RayGUICPP::ProgressBar	208
ProtocolParser	209
RayGui	218
RayGUICPP::RayGui	227
RayLib	228
ZappyTypes::Rectangle	242
Renderer	243
raylibcpp::RenderState	246
resource_dist_s	247
ResourceData	248
SystemWrapper::SafeBuffer	250
SystemWrapper::SafePollFd	251
SystemWrapper::SafeSockAddr	253
RayGUICPP::ScrollPane	255

server_s	
Structure principale du serveur Zappy	256
ServerMessageData	259
raylibcpp::Shape	260
Zappy::SimpleTileRenderStrategy	262
RayGUICPP::Slider	264
socketManager.SocketManager	264
raylibcpp::SoundWrap	267
RayGUICPP::Spinner	269
RayGUICPP::StatusBar	269
RayGUICPP::TabBar	270
TcpConnection	270
TeamNameData	273
raylibcpp::Text3D	274
raylibcpp::Text3DCodepoint	276
raylibcpp::Text3DConfig	277
raylibcpp::Text3DMeasurement	280
raylibcpp::Text3DParser	281
raylibcpp::Text3DRenderer	282
raylibcpp::Text3DWave	283
RayGUICPP::TextBox	284
RayGUICPP::TextInputBox	285
raylibcpp::Texture	286
raylibcpp::Texture3DWrap	287
raylibcpp::TextureCoordinates	288
TextureManager	
Gestionnaire de textures singleton qui implémente un pattern de pool de ressources pour éviter de charger plusieurs fois les mêmes textures	289
tile_s	293
TileContentData	294
TileData	
Structure représentant une tuile dans le contexte graphique	298
Zappy::TileRenderStrategyFactory	299
TimeUnitData	300
RayGUICPP::Toggle	302
RayGUICPP::ToggleGroup	302
UIRenderer	303
behaviors.UpgradeBehavior	305
raylibcpp::Utils	306
RayGUICPP::ValueBox	307
ZappyTypes::Vector2	307
ZappyTypes::Vector3	308
raylibcpp::Vector3Wrap	309
raylibcpp::WaveTextConfig	311
raylibcpp::Window	311
RayGUICPP::WindowBox	313

Chapter 11

File Index

11.1 File List

Here is a list of all files with brief descriptions:

/root/Desktop/Zappy/libs/ TypeAdapter.hpp	351
/root/Desktop/Zappy/libs/RayGUICPP/ RayGui.cpp	317
/root/Desktop/Zappy/libs/RayGUICPP/ RayGui.hpp	317
/root/Desktop/Zappy/libs/RayGUICPP/ raygui_impl.cpp	318
/root/Desktop/Zappy/libs/RayGUICPP/include/ RayGuiEncap.hpp	315
/root/Desktop/Zappy/libs/RaylibCPP/ RayLib.cpp	342
/root/Desktop/Zappy/libs/RaylibCPP/ RayLib.hpp	343
/root/Desktop/Zappy/libs/RaylibCPP/audio/ Audio.cpp	320
/root/Desktop/Zappy/libs/RaylibCPP/audio/ Audio.hpp	320
/root/Desktop/Zappy/libs/RaylibCPP/camera/ Camera.cpp	322
/root/Desktop/Zappy/libs/RaylibCPP/camera/ Camera.hpp	322
/root/Desktop/Zappy/libs/RaylibCPP/font/ Font.cpp	323
/root/Desktop/Zappy/libs/RaylibCPP/font/ Font.hpp	324
/root/Desktop/Zappy/libs/RaylibCPP/font/ Text3D.cpp	325
/root/Desktop/Zappy/libs/RaylibCPP/font/ Text3D.hpp	326
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/ Text3DCodepoint.cpp	327
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/ Text3DCodepoint.hpp	327
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/ Text3DConfig.cpp	328
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/ Text3DConfig.hpp	329
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/ Text3DCore.hpp	330
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/ Text3DHelper.cpp	332
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/ Text3DHelper.hpp	333
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/ Text3DMeasurement.cpp	334
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/ Text3DMeasurement.hpp	335
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/ Text3DParser.cpp	335
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/ Text3DParser.hpp	336
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/ Text3DRenderer.cpp	337
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/ Text3DRenderer.hpp	338
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/ Text3DWave.cpp	339
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/ Text3DWave.hpp	339
/root/Desktop/Zappy/libs/RaylibCPP/input/ Input.cpp	340
/root/Desktop/Zappy/libs/RaylibCPP/input/ Input.hpp	341
/root/Desktop/Zappy/libs/RaylibCPP/shape/ Shape.cpp	344
/root/Desktop/Zappy/libs/RaylibCPP/shape/ Shape.hpp	344
/root/Desktop/Zappy/libs/RaylibCPP/texture/ Texture.cpp	345
/root/Desktop/Zappy/libs/RaylibCPP/texture/ Texture.hpp	346
/root/Desktop/Zappy/libs/RaylibCPP/utils/ Utils.cpp	347
/root/Desktop/Zappy/libs/RaylibCPP/utils/ Utils.hpp	348
/root/Desktop/Zappy/libs/RaylibCPP/window/ Window.cpp	349
/root/Desktop/Zappy/libs/RaylibCPP/window/ Window.hpp	350

/root/Desktop/Zappy/src/AI/main.py	353
/root/Desktop/Zappy/src/AI/agent/actions.py	352
/root/Desktop/Zappy/src/AI/agent/agent.py	352
/root/Desktop/Zappy/src/AI/agent/behaviors.py	352
/root/Desktop/Zappy/src/AI/agent/broadcastManager.py	352
/root/Desktop/Zappy/src/AI/agent/decisionManager.py	352
/root/Desktop/Zappy/src/AI/agent/socketManager.py	353
/root/Desktop/Zappy/src/AI/constants/resources.py	353
/root/Desktop/Zappy/src/AI/constants/roles.py	353
/root/Desktop/Zappy/src/AI/constants/upgrades.py	353
/root/Desktop/Zappy/src/AI/logger/logger.py	353
/root/Desktop/Zappy/src/AI/parser/paringArgsClass.py	354
/root/Desktop/Zappy/src/AI/utils/encryption.py	354
/root/Desktop/Zappy/src/AI/utils/input.py	354
/root/Desktop/Zappy/src/AI/utils/zappy.py	355
/root/Desktop/Zappy/src/GUI/Constants.hpp	356
/root/Desktop/Zappy/src/GUI/GameLoop.cpp	360
/root/Desktop/Zappy/src/GUI/GameLoop.hpp	360
/root/Desktop/Zappy/src/GUI/main.cpp	363
/root/Desktop/Zappy/src/GUI/cameraController/CameraController.cpp	355
/root/Desktop/Zappy/src/GUI/cameraController/CameraController.hpp	355
/root/Desktop/Zappy/src/GUI/graphicalContext/GraphicalContext.cpp	361
/root/Desktop/Zappy/src/GUI/graphicalContext/GraphicalContext.hpp	361
/root/Desktop/Zappy/src/GUI/network/buffer/MessageQueue.cpp	364
/root/Desktop/Zappy/src/GUI/network/buffer/MessageQueue.hpp	364
/root/Desktop/Zappy/src/GUI/network/connection/TcpConnection.cpp	365
/root/Desktop/Zappy/src/GUI/network/connection/TcpConnection.hpp	366
/root/Desktop/Zappy/src/GUI/network/networkManager/NetworkLogger.hpp	367
/root/Desktop/Zappy/src/GUI/network/networkManager/NetworkManager.cpp	367
/root/Desktop/Zappy/src/GUI/network/networkManager/NetworkManager.hpp	368
/root/Desktop/Zappy/src/GUI/network/networkThread/NetworkThread.cpp	369
/root/Desktop/Zappy/src/GUI/network/networkThread/NetworkThread.hpp	369
/root/Desktop/Zappy/src/GUI/network/protocol/HeaderMessage.hpp	370
/root/Desktop/Zappy/src/GUI/network/protocol/Message.cpp	373
/root/Desktop/Zappy/src/GUI/network/protocol/Message.hpp	374
/root/Desktop/Zappy/src/GUI/network/protocol/ProtocolParser.cpp	389
/root/Desktop/Zappy/src/GUI/network/protocol/ProtocolParser.hpp	389
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/BroadcastData.hpp	374
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/EggData.hpp	375
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/EndGameData.hpp	376
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/IMessageData.hpp	377
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/IncantationData.hpp	378
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/IncantationEndData.hpp	379
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/MapSizeData.hpp	380
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/MessageDataAll.hpp	381
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/PlayerInfoData.hpp	382
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/PlayerInventoryData.hpp	383
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/ResourceData.hpp	384
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/ServerMessageData.hpp	385
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/TeamNameData.hpp	386
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/TileContentData.hpp	387
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/TimeUnitData.hpp	388
/root/Desktop/Zappy/src/GUI/network/utils/SystemWrapper.cpp	390
/root/Desktop/Zappy/src/GUI/network/utils/SystemWrapper.hpp	391
/root/Desktop/Zappy/src/GUI/parsing/ParsingCLI.cpp	393
/root/Desktop/Zappy/src/GUI/parsing/ParsingCLI.hpp	393
/root/Desktop/Zappy/src/GUI/renderer/MapRenderer.cpp	395
/root/Desktop/Zappy/src/GUI/renderer/MapRenderer.hpp	395

/root/Desktop/Zappy/src/GUI/renderer/MapRendererObserver.cpp	396
/root/Desktop/Zappy/src/GUI/renderer/MapRendererObserver.hpp	397
/root/Desktop/Zappy/src/GUI/renderer/Renderer.cpp	397
/root/Desktop/Zappy/src/GUI/renderer/Renderer.hpp	398
/root/Desktop/Zappy/src/GUI/renderer/UIRenderer.cpp	405
/root/Desktop/Zappy/src/GUI/renderer/UIRenderer.hpp	406
/root/Desktop/Zappy/src/GUI/renderer/strategies/DetailedTileRenderStrategy.cpp	399
/root/Desktop/Zappy/src/GUI/renderer/strategies/DetailedTileRenderStrategy.hpp	399
/root/Desktop/Zappy/src/GUI/renderer/strategies/ITileRenderStrategy.hpp	400
/root/Desktop/Zappy/src/GUI/renderer/strategies/ModelTileRenderStrategy.cpp	401
/root/Desktop/Zappy/src/GUI/renderer/strategies/ModelTileRenderStrategy.hpp	401
/root/Desktop/Zappy/src/GUI/renderer/strategies/SimpleTileRenderStrategy.cpp	402
/root/Desktop/Zappy/src/GUI/renderer/strategies/SimpleTileRenderStrategy.hpp	403
/root/Desktop/Zappy/src/GUI/renderer/strategies/TileRenderStrategyFactory.cpp	404
/root/Desktop/Zappy/src/GUI/renderer/strategies/TileRenderStrategyFactory.hpp	404
/root/Desktop/Zappy/src/shared/exception/AException.cpp	407
/root/Desktop/Zappy/src/shared/exception/AException.hpp	408
/root/Desktop/Zappy/src/shared/exception/IException.hpp	408
/root/Desktop/Zappy/src/textureManager/ModelManager.cpp	409
/root/Desktop/Zappy/src/textureManager/ModelManager.hpp	410
/root/Desktop/Zappy/src/textureManager/ModelManagerAdapter.hpp	411
/root/Desktop/Zappy/src/textureManager/TextureManager.cpp	412
/root/Desktop/Zappy/src/textureManager/TextureManager.hpp	413
/root/Desktop/Zappy/src/Server/free.c	455
/root/Desktop/Zappy/src/Server/help.c	489
/root/Desktop/Zappy/src/Server/main.c	544
/root/Desktop/Zappy/src/Server/parsing.c	578
/root/Desktop/Zappy/src/Server/parsing_err.c	582
/root/Desktop/Zappy/src/Server/buffer/circular_buffer.c	414
/root/Desktop/Zappy/src/Server/command/broadcast.c	416
/root/Desktop/Zappy/src/Server/command/connect_nbr.c	419
/root/Desktop/Zappy/src/Server/command/eject.c	421
/root/Desktop/Zappy/src/Server/command/finish_incantation.c	425
/root/Desktop/Zappy/src/Server/command/fork.c	427
/root/Desktop/Zappy/src/Server/command/forward.c	429
/root/Desktop/Zappy/src/Server/command/inventory.c	430
/root/Desktop/Zappy/src/Server/command/left.c	431
/root/Desktop/Zappy/src/Server/command/look.c	433
/root/Desktop/Zappy/src/Server/command/look_utils.c	435
/root/Desktop/Zappy/src/Server/command/parse_command.c	438
/root/Desktop/Zappy/src/Server/command/parse_command_utils.c	441
/root/Desktop/Zappy/src/Server/command/parse_command_utils_bis.c	443
/root/Desktop/Zappy/src/Server/command/pending_cmd_utils.c	445
/root/Desktop/Zappy/src/Server/command/right.c	446
/root/Desktop/Zappy/src/Server/command/set_object.c	447
/root/Desktop/Zappy/src/Server/command/start_incantation.c	448
/root/Desktop/Zappy/src/Server/command/take_object.c	451
/root/Desktop/Zappy/src/Server/command/tile_to_str.c	453
/root/Desktop/Zappy/src/Server/graphical_command/command_bct.c	458
/root/Desktop/Zappy/src/Server/graphical_command/command_ebo.c	461
/root/Desktop/Zappy/src/Server/graphical_command/command_edi.c	462
/root/Desktop/Zappy/src/Server/graphical_command/command_enw.c	464
/root/Desktop/Zappy/src/Server/graphical_command/command_msz.c	466
/root/Desktop/Zappy/src/Server/graphical_command/command_pbc.c	467
/root/Desktop/Zappy/src/Server/graphical_command/command_pdi.c	468
/root/Desktop/Zappy/src/Server/graphical_command/command_pdr.c	469
/root/Desktop/Zappy/src/Server/graphical_command/command_pex.c	470
/root/Desktop/Zappy/src/Server/graphical_command/command_pfk.c	471

/root/Desktop/Zappy/src/Server/graphical_command/command_pgt.c	472
/root/Desktop/Zappy/src/Server/graphical_command/command_pic.c	473
/root/Desktop/Zappy/src/Server/graphical_command/command_pie.c	474
/root/Desktop/Zappy/src/Server/graphical_command/command_pin.c	476
/root/Desktop/Zappy/src/Server/graphical_command/command_plv.c	478
/root/Desktop/Zappy/src/Server/graphical_command/command_pnw.c	480
/root/Desktop/Zappy/src/Server/graphical_command/command_ppo.c	482
/root/Desktop/Zappy/src/Server/graphical_command/command_seg.c	483
/root/Desktop/Zappy/src/Server/graphical_command/command_sgt.c	485
/root/Desktop/Zappy/src/Server/graphical_command/command_smg.c	486
/root/Desktop/Zappy/src/Server/graphical_command/command_sst.c	487
/root/Desktop/Zappy/src/Server/graphical_command/command_tna.c	488
/root/Desktop/Zappy/src/Server/include/circular_buffer.h	491
/root/Desktop/Zappy/src/Server/include/client.h	493
/root/Desktop/Zappy/src/Server/include/command.h	495
/root/Desktop/Zappy/src/Server/include/egg.h	514
/root/Desktop/Zappy/src/Server/include/graphical_commands.h	515
/root/Desktop/Zappy/src/Server/include/parsing.h	522
/root/Desktop/Zappy/src/Server/include/pending_cmd.h	524
/root/Desktop/Zappy/src/Server/include/pending_cmd_utils.h	525
/root/Desktop/Zappy/src/Server/include/player.h	527
/root/Desktop/Zappy/src/Server/include/server.h	531
/root/Desktop/Zappy/src/Server/include/tile.h	540
/root/Desktop/Zappy/src/Server/map/egg.c	548
/root/Desktop/Zappy/src/Server/map/map_creation.c	551
/root/Desktop/Zappy/src/Server/map/resources_map.c	552
/root/Desktop/Zappy/src/Server/map/tick_update.c	554
/root/Desktop/Zappy/src/Server/map/tile.c	556
/root/Desktop/Zappy/src/Server/network/array_function.c	558
/root/Desktop/Zappy/src/Server/network/client_node.c	561
/root/Desktop/Zappy/src/Server/network/connection.c	563
/root/Desktop/Zappy/src/Server/network/connection_err.c	567
/root/Desktop/Zappy/src/Server/network/connection_utils.c	569
/root/Desktop/Zappy/src/Server/network/graphical_client.c	570
/root/Desktop/Zappy/src/Server/network/queue_connection.c	573
/root/Desktop/Zappy/src/Server/network/server_run.c	575
/root/Desktop/Zappy/src/Server/player/player_death.c	583
/root/Desktop/Zappy/src/Server/player/player_init.c	585
/root/Desktop/Zappy/src/Server/player/player_inventory.c	587
/root/Desktop/Zappy/src/Server/player/player_inventory_display.c	589
/root/Desktop/Zappy/src/Server/player/player_inventory_utils.c	591
/root/Desktop/Zappy/src/Shared/Common.hpp	592
/root/Desktop/Zappy/src/Shared/DLLoader.hpp	594
/root/Desktop/Zappy/src/Shared/IGraphicsLib.hpp	595
/root/Desktop/Zappy/src/Shared/IGuiLib.hpp	596
/root/Desktop/Zappy/src/Shared/LibraryManager.hpp	597

Chapter 12

Namespace Documentation

12.1 actions Namespace Reference

Functions

- def `go_to_pos_with_distance` (agent, distance)
- def `take_all_of_item_here` (agent, item)
- def `take_everything_here` (agent)
- def `go_take_item` (agent, item)
- def `got_to_dir` (agent, direction)

12.1.1 Function Documentation

12.1.1.1 `go_take_item()`

```
def actions.go_take_item (
    agent,
    item )
```

Definition at line 46 of file actions.py.

```
46 def go_take_item(agent, item):
47     nb_turns = 0
48     i = 0
49
50     while True:
51         surroundings = agent.send_command("Look")
52         if surroundings is None or "ko" in surroundings:
53             print(f"go_take_item: Failed to look around. Response: {surroundings}")
54             return
55         closest_item_distance = zappy.get_closest_of_item(surroundings, item)
56         if closest_item_distance != -1 or i > 4:
57             break
58         agent.send_command("Right")
59         nb_turns += 1
60         if nb_turns > 3:
61             i += 1
62             for _ in range(i * agent.level):
63                 agent.send_command("Forward")
64             nb_turns = 0
65
66         if closest_item_distance == 0:
67             take_all_of_item_here(agent, item)
68             return
69
70         distance_to_item = zappy.get_closest_of_item(surroundings, item)
71
72         go_to_pos_with_distance(agent, distance_to_item)
73         take_all_of_item_here(agent, item)
74
```

12.1.1.2 go_to_pos_with_distance()

```
def actions.go_to_pos_with_distance (
    agent,
    distance )
Definition at line 4 of file actions.py.
4 def go_to_pos_with_distance(agent, distance):
5     if (distance == 0):
6         return
7
8     xrelative, yrelative = zappy.get_item_relative_pos(distance)
9     for _ in range(yrelative):
10        agent.send_command("Forward")
11
12    if xrelative > 0:
13        agent.send_command("Right")
14    elif xrelative < 0:
15        agent.send_command("Left")
16
17    for _ in range(abs(xrelative)):
18        agent.send_command("Forward")
19
```

12.1.1.3 got_to_dir()

```
def actions.got_to_dir (
    agent,
    direction )
```

Definition at line 75 of file actions.py.

```
75 def got_to_dir(agent, direction):
76     if direction == 1 or direction == 2:
77         agent.send_command("Forward")
78     elif direction == 3 or direction == 4:
79         agent.send_command("Left")
80         agent.send_command("Forward")
81     elif direction == 5 or direction == 6:
82         agent.send_command("Left")
83         agent.send_command("Left")
84         agent.send_command("Forward")
85     elif direction == 7 or direction == 8:
86         agent.send_command("Right")
87         agent.send_command("Forward")
```

12.1.1.4 take_all_of_item_here()

```
def actions.take_all_of_item_here (
    agent,
    item )
```

Definition at line 20 of file actions.py.

```
20 def take_all_of_item_here(agent, item):
21     surroundings = agent.send_command("Look")
22     while zappy.get_closest_of_item(surroundings, item) == 0:
23         if surroundings is None or "ko" in surroundings:
24             print(f"take_all_of_item_here: Failed to look around. Response: {surroundings}")
25             return
26         response = agent.send_command("Take " + item)
27         if (response is None or "ko" in response):
28             print(f"Failed to take {item}. Response: {response}")
29             break
30     surroundings = agent.send_command("Look")
31
```

12.1.1.5 take_everything_here()

```
def actions.take_everything_here (
    agent )
```

Definition at line 32 of file actions.py.

```
32 def take_everything_here(agent):
33     surroundings = agent.send_command("Look")
34     if surroundings is None or "ko" in surroundings:
35         print(f"take_everything_here: Failed to look around. Response: {surroundings}")
```

```
36     return
37
38     items_on_ground = surroundings.strip(" [ ]").split(",") [0].split(" ")
39
40     for item in items_on_ground:
41         if item:
42             item = item.strip(", .")
43             if item != "player":
44                 take_all_of_item_here(agent, item)
45
```

12.2 agent Namespace Reference

Data Structures

- class [Agent](#)

12.3 behaviors Namespace Reference

Data Structures

- class [Behavior](#)
- class [GetFoodBehavior](#)
- class [UpgradeBehavior](#)
- class [GetMineralsBehavior](#)
- class [DysonBehavior](#)
- class [GetFoodAndMineralsBehavior](#)

12.4 broadcastManager Namespace Reference

Data Structures

- class [BroadcastManager](#)

12.5 decisionManager Namespace Reference

Data Structures

- class [DecisionManager](#)

12.6 encryption Namespace Reference

Functions

- def [encrypt_message](#) (message)
- def [decrypt_message](#) (encrypted_message)

Variables

- [secret_key](#) = os.getenv("SECRET_KEY", "default_secret_key")

12.6.1 Function Documentation

12.6.1.1 decrypt_message()

```
def encryption.decrypt_message (
    encrypted_message )
```

Definition at line 29 of file encryption.py.

```
29 def decrypt_message(encrypted_message):
30     try:
31         fernet = Fernet(_get_encryption_key())
32         encrypted_data = base64.urlsafe_b64decode(encrypted_message.encode())
33         decrypted_data = fernet.decrypt(encrypted_data)
34         return decrypted_data.decode()
35
36     except Exception as e:
37         print(f"Decryption error: {e}, with message: {encrypted_message}")
38     return None
```

12.6.1.2 encrypt_message()

```
def encryption.encrypt_message (
    message )
```

Definition at line 19 of file encryption.py.

```
19 def encrypt_message(message):
20     try:
21         fernet = Fernet(_get_encryption_key())
22         encrypted_data = fernet.encrypt(message.encode())
23         return base64.urlsafe_b64encode(encrypted_data).decode()
24
25     except Exception as e:
26         print(f"Encryption error: {e}")
27     return message
28
```

12.6.2 Variable Documentation

12.6.2.1 secret_key

```
encryption.secret_key = os.getenv("SECRET_KEY", "default_secret_key")
```

Definition at line 12 of file encryption.py.

12.7 input Namespace Reference

Functions

- def [get_key \(\)](#)

12.7.1 Function Documentation

12.7.1.1 get_key()

```
def input.get_key ( )
```

Definition at line 6 of file input.py.

```
6 def get_key():
7     fd = sys.stdin.fileno()
8     old_settings = termios.tcgetattr(fd)
9     try:
10         tty.setraw(sys.stdin.fileno())
11         if select.select([sys.stdin], [], [], 0.1) == ([sys.stdin], [], []):
12             key = sys.stdin.read(1)
13             if key == '\x1b':
14                 key += sys.stdin.read(2)
15     return key
16 finally:
17     termios.tcsetattr(fd, termios.TCSADRAIN, old_settings)
18 return None
```

12.8 logger Namespace Reference

Data Structures

- class [LogLevel](#)
- class [Logger](#)

12.9 main Namespace Reference

Functions

- def [run_agent](#) (ip, port, team, agent_id)
- def [signal_handler](#) (sig, frame)

Variables

- [args](#) = parseArgs()
- int [num_agents](#) = 1
- [port](#) = args.getPort()
- [ip](#) = args.getMachine()
- [team](#) = args.getName()
- list [child_pids](#) = []
- [pid](#) = os.fork()

12.9.1 Function Documentation

12.9.1.1 run_agent()

```
def main.run_agent (
    ip,
    port,
    team,
    agent_id )
```

Definition at line 10 of file main.py.

```
10 def run_agent(ip, port, team, agent_id):
11     try:
12         ai = Agent(ip, port, team, agent_id)
13         ai.start()
14     except Exception as e:
15         print(f"Agent {agent_id} failed: {e}")
16         sys.exit(1)
17
```

12.9.1.2 signal_handler()

```
def main.signal_handler (
    sig,
    frame )
```

Definition at line 18 of file main.py.

```
18 def signal_handler(sig, frame):
19     for pid in child_pids:
20         try:
21             os.kill(pid, signal.SIGTERM)
22         except ProcessLookupError:
23             pass
24     sys.exit(0)
25
```

12.9.2 Variable Documentation

12.9.2.1 args

```
main.args = parseArgs()  
Definition at line 27 of file main.py.
```

12.9.2.2 child_pids

```
list main.child_pids = []  
Definition at line 35 of file main.py.
```

12.9.2.3 ip

```
main.ip = args.getMachine()  
Definition at line 30 of file main.py.
```

12.9.2.4 num_agents

```
int main.num_agents = 1  
Definition at line 28 of file main.py.
```

12.9.2.5 pid

```
main.pid = os.fork()  
Definition at line 41 of file main.py.
```

12.9.2.6 port

```
main.port = args.getPort()  
Definition at line 29 of file main.py.
```

12.9.2.7 team

```
main.team = args.getName()  
Definition at line 31 of file main.py.
```

12.10 paringArgsClass Namespace Reference

Data Structures

- class [ParsingArgsError](#)
- class [parseArgs](#)

12.11 RayGUICPP Namespace Reference

Data Structures

- class [RayGui](#)
- class [Button](#)
- class [Label](#)
- class [CheckBox](#)
- class [Toggle](#)
- class [ToggleGroup](#)

- class [ComboBox](#)
- class [DropdownBox](#)
- class [TextBox](#)
- class [ValueBox](#)
- class [Spinner](#)
- class [Slider](#)
- class [ProgressBar](#)
- class [StatusBar](#)
- class [WindowBox](#)
- class [GroupBox](#)
- class [Panel](#)
- class [ScrollPane](#)
- class [TabBar](#)
- class [ListView](#)
- class [ColorPicker](#)
- class [MessageBox](#)
- class [TextInputBox](#)
- class [Icon](#)

12.12 raylibcpp Namespace Reference

Namespaces

- [Text3DHelper](#)

Data Structures

- class [Audio](#)
- class [SoundWrap](#)
- class [MusicWrap](#)
- class [Camera2DWrap](#)
- class [Camera3DWrap](#)
- class [Font](#)
- class [Text3DCodepoint](#)
- class [Text3DConfig](#)
- struct [WaveTextConfig](#)
- struct [GlyphDimensions](#)
- struct [TextureCoordinates](#)
- struct [RenderState](#)
- class [Text3DMeasurement](#)
- class [Text3DParser](#)
- class [Text3DRenderer](#)
- class [Text3DWave](#)
- class [Text3D](#)
- class [Input](#)
- class [Shape](#)
- class [Texture](#)
- class [Texture3DWrap](#)
- class [ModelWrap](#)
- class [Utils](#)
- class [Vector3Wrap](#)
- class [Window](#)

Typedefs

- using `WaveTextConfigPtr` = std::shared_ptr<`WaveTextConfig`>
- using `WaveTextConfigUPtr` = std::unique_ptr<`WaveTextConfig`>
- using `RenderStatePtr` = std::shared_ptr<`RenderState`>

12.12.1 Typedef Documentation

12.12.1.1 RenderStatePtr

```
using raylibcpp::RenderStatePtr = typedef std::shared_ptr<RenderState>
Definition at line 31 of file Text3DHelper.hpp.
```

12.12.1.2 WaveTextConfigPtr

```
using raylibcpp::WaveTextConfigPtr = typedef std::shared_ptr<WaveTextConfig>
Definition at line 25 of file Text3DCore.hpp.
```

12.12.1.3 WaveTextConfigUPtr

```
using raylibcpp::WaveTextConfigUPtr = typedef std::unique_ptr<WaveTextConfig>
Definition at line 26 of file Text3DCore.hpp.
```

12.13 raylibcpp::Text3DHelper Namespace Reference

Functions

- int `getGlyphIndex` (const ::Font &font, int codepoint)
- float `calculateScale` (const ::Font &font, float fontSize)
- Vector3 `calculateGlyphPosition` (const ::Font &font, int index, Vector3 basePosition, float scale)
- Rectangle `calculateSourceRectangle` (const ::Font &font, int index)
- GlyphDimensions `calculateGlyphDimensions` (const ::Font &font, int index, float scale)
- bool `isRenderableCharacter` (int codepoint)
- float `calculateAdvanceX` (const ::Font &font, int index, float scale, float fontSpacing)
- TextureCoordinates `calculateTextureCoordinates` (const ::Font &font, Rectangle srcRec)
- bool `isValidFont` (const ::Font &font)
- bool `isValidRenderParams` (float fontSize, float fontSpacing, float lineSpacing)
- bool `isWhitespace` (int codepoint)
- bool `isNewline` (int codepoint)

Variables

- static constexpr float `MIN_FONT_SIZE` = 0.1f
- static constexpr float `MAX_FONT_SIZE` = 1000.0f
- static constexpr float `MIN_SPACING` = -100.0f
- static constexpr float `MAX_SPACING` = 1000.0f

12.13.1 Function Documentation

12.13.1.1 calculateAdvanceX()

```
float raylibcpp::Text3DHelper::calculateAdvanceX (
    const ::Font & font,
    int index,
    float scale,
    float fontSpacing )
```

Definition at line 50 of file Text3DHelper.cpp.

```
50
51     if (font.glyphs[index].advanceX == 0) {
52         return static_cast<float>(font.recs[index].width) * scale + fontSpacing;
53     }
54     return static_cast<float>(font.glyphs[index].advanceX) * scale + fontSpacing;
55 }
```

12.13.1.2 calculateGlyphDimensions()

```
GlyphDimensions raylibcpp::Text3DHelper::calculateGlyphDimensions (
    const ::Font & font,
    int index,
    float scale )
```

Definition at line 39 of file Text3DHelper.cpp.

```
39
40     return {
41         static_cast<float>(font.recs[index].width + 2.0f * font.glyphPadding) * scale,
42         static_cast<float>(font.recs[index].height + 2.0f * font.glyphPadding) * scale
43     };
44 }
```

12.13.1.3 calculateGlyphPosition()

```
Vector3 raylibcpp::Text3DHelper::calculateGlyphPosition (
    const ::Font & font,
    int index,
    Vector3 basePosition,
    float scale )
```

Definition at line 23 of file Text3DHelper.cpp.

```
23
24     Vector3 position = basePosition;
25     position.x += static_cast<float>(font.glyphs[index].offsetX - font.glyphPadding) * scale;
26     position.z += static_cast<float>(font.glyphs[index].offsetY - font.glyphPadding) * scale;
27     return position;
28 }
```

12.13.1.4 calculateScale()

```
float raylibcpp::Text3DHelper::calculateScale (
    const ::Font & font,
    float fontSize )
```

Definition at line 19 of file Text3DHelper.cpp.

```
19
20     return fontSize / static_cast<float>(font.baseSize);
21 }
```

12.13.1.5 calculateSourceRectangle()

```
Rectangle raylibcpp::Text3DHelper::calculateSourceRectangle (
    const ::Font & font,
    int index )
```

Definition at line 30 of file Text3DHelper.cpp.

```
30
31     return {
32         font.recs[index].x - static_cast<float>(font.glyphPadding),
33         font.recs[index].y - static_cast<float>(font.glyphPadding),
```

```

34         font.recs[index].width + 2.0f * font.glyphPadding,
35         font.recs[index].height + 2.0f * font.glyphPadding
36     };
37 }

```

12.13.1.6 calculateTextureCoordinates()

```
TextureCoordinates raylibcpp::Text3DHelper::calculateTextureCoordinates (
    const ::Font & font,
    Rectangle srcRec )
```

Definition at line 57 of file Text3DHelper.cpp.

```

57
58     return {
59         srcRec.x / font.texture.width,
60         srcRec.y / font.texture.height,
61         (srcRec.x + srcRec.width) / font.texture.width,
62         (srcRec.y + srcRec.height) / font.texture.height
63     };
64 }
```

12.13.1.7 getGlyphIndex()

```
int raylibcpp::Text3DHelper::getGlyphIndex (
    const ::Font & font,
    int codepoint )
```

Definition at line 15 of file Text3DHelper.cpp.

```

15
16     return GetGlyphIndex(font, codepoint);
17 }
```

12.13.1.8 isNewline()

```
bool raylibcpp::Text3DHelper::isNewline (
    int codepoint )
```

Definition at line 80 of file Text3DHelper.cpp.

```

80
81     return codepoint == '\n';
82 }
```

12.13.1.9 isRenderableCharacter()

```
bool raylibcpp::Text3DHelper::isRenderableCharacter (
    int codepoint )
```

Definition at line 46 of file Text3DHelper.cpp.

```

46
47     return (codepoint != ' ') && (codepoint != '\t');
48 }
```

12.13.1.10 isValidFont()

```
bool raylibcpp::Text3DHelper::isValidFont (
    const ::Font & font )
```

Definition at line 66 of file Text3DHelper.cpp.

```

66
67     return font.texture.id > 0 && font.baseSize > 0 && font.glyphs != nullptr && font.recs != nullptr;
68 }
```

12.13.1.11 isValidRenderParams()

```
bool raylibcpp::Text3DHelper::isValidRenderParams (
    float fontSize,
```

```

        float fontSpacing,
        float lineSpacing )
Definition at line 70 of file Text3DHelper.cpp.
70
71     return fontSize >= MIN_FONT_SIZE && fontSize <= MAX_FONT_SIZE &&
72         fontSpacing >= MIN_SPACING && fontSpacing <= MAX_SPACING &&
73         lineSpacing >= MIN_SPACING && lineSpacing <= MAX_SPACING;
74 }
```

12.13.1.12 isWhitespace()

```

bool raylibcpp::Text3DHelper::isWhitespace (
    int codepoint )
Definition at line 76 of file Text3DHelper.cpp.
76
77     return codepoint == ' ' || codepoint == '\t';
78 }
```

12.13.2 Variable Documentation

12.13.2.1 MAX_FONT_SIZE

```
constexpr float raylibcpp::Text3DHelper::MAX_FONT_SIZE = 1000.0f [static], [constexpr]
Definition at line 48 of file Text3DHelper.hpp.
```

12.13.2.2 MAX_SPACING

```
constexpr float raylibcpp::Text3DHelper::MAX_SPACING = 1000.0f [static], [constexpr]
Definition at line 50 of file Text3DHelper.hpp.
```

12.13.2.3 MIN_FONT_SIZE

```
constexpr float raylibcpp::Text3DHelper::MIN_FONT_SIZE = 0.1f [static], [constexpr]
Definition at line 47 of file Text3DHelper.hpp.
```

12.13.2.4 MIN_SPACING

```
constexpr float raylibcpp::Text3DHelper::MIN_SPACING = -100.0f [static], [constexpr]
Definition at line 49 of file Text3DHelper.hpp.
```

12.14 resources Namespace Reference

Variables

- dictionary [resources](#)

12.14.1 Variable Documentation

12.14.1.1 resources

```
dictionary resources.resources
```

Initial value:

```
1 = {
2     "food": 0,
```

```

3     "linemate": 1,
4     "deraumere": 2,
5     "sibur": 3,
6     "mendiane": 4,
7     "phiras": 5,
8     "thystame": 6
9 }

```

Definition at line 2 of file resources.py.

12.15 roles Namespace Reference

Variables

- dictionary [roles](#)

12.15.1 Variable Documentation

12.15.1.1 roles

```
dictionary roles.roles
Initial value:
1 = {
2     "captain": {
3         "minimum_amount": 1,
4         "maximum_amount": 1,
5     },
6     "attacker": {
7         "minimum_amount": 0,
8         "maximum_amount": 0,
9     },
10    "miner": {
11        "minimum_amount": 1,
12        "maximum_amount": 2,
13    },
14
15 }
```

Definition at line 2 of file roles.py.

12.16 socketManager Namespace Reference

Data Structures

- class [SocketManager](#)

12.17 SystemWrapper Namespace Reference

Data Structures

- class [SafeSockAddr](#)
- class [SafePollFd](#)
- class [SafeBuffer](#)

Functions

- int [createSocket](#) (int domain, int type, int protocol)
- int [connectSocket](#) (int sockfd, const [SafeSockAddr](#) &addr)
- int [closeSocket](#) (int fd)
- int [setNonBlocking](#) (int fd)
- ssize_t [readSocket](#) (int fd, std::shared_ptr< [SafeBuffer](#) > buffer, size_t count)
- ssize_t [writeSocket](#) (int fd, const [SafeBuffer](#) &buffer, size_t count)
- int [pollSocket](#) (const [SafePollFd](#) &pfd, int timeout)

- int `pollSocket` (const std::vector< `SafePollFd` > &pollfds, int timeout)
- bool `inetPton` (int af, const std::string &src, void *dst)
- std::string `inetNtop` (int af, const void *src)
- std::string `getErrorResponse` ()
- void `executeWithTimeout` (const std::function< bool()> &action, int timeoutMs)
- template<typename T>
int `getSocketOption` (int sockfd, int level, int optname, T *optval)
- template<typename T>
std::unique_ptr< T[]> `makeUniqueArray` (size_t size)
- template<typename T>
std::shared_ptr< T[]> `makeSharedArray` (size_t size)

12.17.1 Function Documentation

12.17.1.1 `closeSocket()`

```
int SystemWrapper::closeSocket (
    int fd )
Definition at line 100 of file SystemWrapper.cpp.
100
101     return close(fd);
102 }
```

12.17.1.2 `connectSocket()`

```
int SystemWrapper::connectSocket (
    int sockfd,
    const SafeSockAddr & addr )
Definition at line 96 of file SystemWrapper.cpp.
96
97     return connect(sockfd, addr.getSockAddr(), addr.getSize());
98 }
```

12.17.1.3 `createSocket()`

```
int SystemWrapper::createSocket (
    int domain,
    int type,
    int protocol )
Definition at line 92 of file SystemWrapper.cpp.
92
93     return socket(domain, type, protocol);
94 }
```

12.17.1.4 `executeWithTimeout()`

```
void SystemWrapper::executeWithTimeout (
    const std::function< bool()> & action,
    int timeoutMs )
Definition at line 189 of file SystemWrapper.cpp.
189
190     auto start = std::chrono::steady_clock::now();
191     while (true) {
192         if (action())
193             return;
194         auto elapsed = std::chrono::steady_clock::now() - start;
195         auto elapsedMs = std::chrono::duration_cast<std::chrono::milliseconds>(elapsed).count();
196         if (elapsedMs >= timeoutMs) {
197             return;
198     }
```

```

199         std::this_thread::sleep_for(std::chrono::milliseconds(1));
200     }
201 }
```

12.17.1.5 getErrorString()

`std::string SystemWrapper::getErrorString ()`

Definition at line 156 of file `SystemWrapper.cpp`.

```

156         {
157     return std::string(strerror(errno));
158 }
```

12.17.1.6 getSocketOption()

```

template<typename T >
int SystemWrapper::getSocketOption (
    int sockfd,
    int level,
    int optname,
    T * optval )
```

Definition at line 108 of file `SystemWrapper.hpp`.

```

108
109     socklen_t optlen = sizeof(T);
110     return getsockopt(sockfd, level, optname, optval, &optlen);
111 }
```

12.17.1.7 inetNtop()

```

std::string SystemWrapper::inetNtop (
    int af,
    const void * src )
```

Definition at line 148 of file `SystemWrapper.cpp`.

```

148
149     char buffer[INET6_ADDRSTRLEN];
150     const char* result = inet_ntop(af, src, buffer, sizeof(buffer));
151     if (result == nullptr)
152         return "";
153     return std::string(buffer);
154 }
```

12.17.1.8 inetPton()

```

bool SystemWrapper::inetPton (
    int af,
    const std::string & src,
    void * dst )
```

Definition at line 144 of file `SystemWrapper.cpp`.

```

144
145     return inet_pton(af, src.c_str(), dst) > 0;
146 }
```

12.17.1.9 makeSharedArray()

```

template<typename T >
std::shared_ptr<T[]> SystemWrapper::makeSharedArray (
    size_t size )
```

Definition at line 127 of file `SystemWrapper.hpp`.

```

127
128     return std::shared_ptr<T[]>(new T[size], std::default_delete<T[]>());
129 }
```

12.17.1.10 makeUniqueArray()

```
template<typename T >
std::unique_ptr<T[]> SystemWrapper::makeUniqueArray (
    size_t size )
Definition at line 122 of file SystemWrapper.hpp.
122
123     return std::make_unique<T[]>(size);
124 }
```

12.17.1.11 pollSocket() [1/2]

```
int SystemWrapper::pollSocket (
    const SafePollFd & pfd,
    int timeout )
Definition at line 131 of file SystemWrapper.cpp.
131
132     return poll(const_cast<pollfd*>(pfd.getPollPtr()), 1, timeout);
133 }
```

12.17.1.12 pollSocket() [2/2]

```
int SystemWrapper::pollSocket (
    const std::vector< SafePollFd > & pollfds,
    int timeout )
Definition at line 135 of file SystemWrapper.cpp.
135
136     std::vector<pollfd> rawPollfds;
137     rawPollfds.reserve(pollfds.size());
138     for (const auto& pfd : pollfds)
139         rawPollfds.push_back(pfd.getPollFd());
140     int result = poll(rawPollfds.data(), rawPollfds.size(), timeout);
141     return result;
142 }
```

12.17.1.13 readSocket()

```
ssize_t SystemWrapper::readSocket (
    int fd,
    std::shared_ptr< SafeBuffer > buffer,
    size_t count )
```

Definition at line 111 of file SystemWrapper.cpp.

```
111
112     if (!buffer)
113         return -1;
114     if (buffer->size() < count) {
115         buffer->resize(count);
116     }
117     std::vector<char> tempBuffer(count + 1, 0);
118     ssize_t bytesRead = read(fd, tempBuffer.data(), count);
119     if (bytesRead > 0) {
120         tempBuffer[bytesRead] = '\0';
121         buffer->data().assign(tempBuffer.data(), bytesRead);
122     }
123     return bytesRead;
124 }
```

12.17.1.14 setNonBlocking()

```
int SystemWrapper::setNonBlocking (
    int fd )
Definition at line 104 of file SystemWrapper.cpp.
104
105     int flags = fcntl(fd, F_GETFL, 0);
106     if (flags == -1)
107         return -1;
```

```
108     return fcntl(fd, F_SETFL, flags | O_NONBLOCK);
109 }
```

12.17.1.15 writeSocket()

```
ssize_t SystemWrapper::writeSocket (
    int fd,
    const SafeBuffer & buffer,
    size_t count )
```

Definition at line 126 of file SystemWrapper.cpp.

```
126
127     const size_t minSize = std::min(count, buffer.size());
128     return write(fd, buffer.data().c_str(), minSize);
129 }
```

12.18 TypeAdapter Namespace Reference

Functions

- inline ::Vector2 ToRaylib (const ZappyTypes::Vector2 &vec)
- ZappyTypes::Vector2 FromRaylib (const ::Vector2 &vec)
- inline ::Vector3 ToRaylib (const ZappyTypes::Vector3 &vec)
- ZappyTypes::Vector3 FromRaylib (const ::Vector3 &vec)
- inline ::Color ToRaylib (const ZappyTypes::Color &color)
- ZappyTypes::Color FromRaylib (const ::Color &color)
- inline ::Rectangle ToRaylib (const ZappyTypes::Rectangle &rect)
- ZappyTypes::Rectangle FromRaylib (const ::Rectangle &rect)

12.18.1 Function Documentation

12.18.1.1 FromRaylib() [1/4]

```
ZappyTypes::Color TypeAdapter::FromRaylib (
    const ::Color & color ) [inline]
```

Definition at line 53 of file TypeAdapter.hpp.

```
53
54     ZappyTypes::Color result;
55     result.r = color.r;
56     result.g = color.g;
57     result.b = color.b;
58     result.a = color.a;
59     return result;
60 }
```

12.18.1.2 FromRaylib() [2/4]

```
ZappyTypes::Rectangle TypeAdapter::FromRaylib (
    const ::Rectangle & rect ) [inline]
```

Definition at line 71 of file TypeAdapter.hpp.

```
71
72     ZappyTypes::Rectangle result;
73     result.x = rect.x;
74     result.y = rect.y;
75     result.width = rect.width;
76     result.height = rect.height;
77     return result;
78 }
```

12.18.1.3 FromRaylib() [3/4]

```
ZappyTypes::Vector2 TypeAdapter::FromRaylib (
    const ::Vector2 & vec) [inline]
```

Definition at line 21 of file TypeAdapter.hpp.

```
21
22     ZappyTypes::Vector2 result;
23     result.x = vec.x;
24     result.y = vec.y;
25     return result;
26 }
```

{

12.18.1.4 FromRaylib() [4/4]

```
ZappyTypes::Vector3 TypeAdapter::FromRaylib (
    const ::Vector3 & vec) [inline]
```

Definition at line 36 of file TypeAdapter.hpp.

```
36
37     ZappyTypes::Vector3 result;
38     result.x = vec.x;
39     result.y = vec.y;
40     result.z = vec.z;
41     return result;
42 }
```

{

12.18.1.5 ToRaylib() [1/4]

```
inline ::Color TypeAdapter::ToRaylib (
    const ZappyTypes::Color & color)
```

Definition at line 44 of file TypeAdapter.hpp.

```
44
45     ::Color result;
46     result.r = color.r;
47     result.g = color.g;
48     result.b = color.b;
49     result.a = color.a;
50     return result;
51 }
```

{

12.18.1.6 ToRaylib() [2/4]

```
inline ::Rectangle TypeAdapter::ToRaylib (
    const ZappyTypes::Rectangle & rect)
```

Definition at line 62 of file TypeAdapter.hpp.

```
62
63     ::Rectangle result;
64     result.x = rect.x;
65     result.y = rect.y;
66     result.width = rect.width;
67     result.height = rect.height;
68     return result;
69 }
```

{

12.18.1.7 ToRaylib() [3/4]

```
inline ::Vector2 TypeAdapter::ToRaylib (
    const ZappyTypes::Vector2 & vec)
```

Definition at line 14 of file TypeAdapter.hpp.

```
14
15     ::Vector2 result;
16     result.x = vec.x;
17     result.y = vec.y;
18     return result;
19 }
```

{

12.18.1.8 ToRaylib() [4/4]

```
inline ::Vector3 TypeAdapter::ToRaylib (
    const ZappyTypes::Vector3 & vec )
Definition at line 28 of file TypeAdapter.hpp.
28
29     ::Vector3 result;
30     result.x = vec.x;
31     result.y = vec.y;
32     result.z = vec.z;
33     return result;
34 }
```

12.19 upgrades Namespace Reference

12.20 zappy Namespace Reference

Functions

- def [get_item_relative_pos](#) (item_position)
- def [parse_inventory](#) (inventory_str)
- def [get_closest_of_item](#) (surroundings_str, item)
- def [go_get_item](#) (surroundings, item)
- def [get_best_available_resource](#) (surroundings)
- def [how_much_of_item_here](#) (surroundings, item)

12.20.1 Function Documentation

12.20.1.1 get_best_available_resource()

```
def zappy.get_best_available_resource (
    surroundings )
```

Definition at line 67 of file zappy.py.

```
67 def get_best_available_resource(surroundings):
68     best_available_resource = None
69     best_priority = 0
70
71     for resource in resources:
72         if get_closest_of_item(surroundings, resource) != -1:
73             resource_priority = resources[resource]
74             if resource_priority > best_priority:
75                 best_priority = resource_priority
76                 best_available_resource = resource
77
78     return best_available_resource
79
80
```

12.20.1.2 get_closest_of_item()

```
def zappy.get_closest_of_item (
    surroundings_str,
    item )
```

Definition at line 43 of file zappy.py.

```
43 def get_closest_of_item(surroundings_str, item):
44     if not surroundings_str:
45         print("get_closest_of_item: Surroundings data is empty or None.")
46         return -1
47
48     cleaned = surroundings_str.strip("[ ]")
49     tiles = cleaned.split(", ")
50
51     for i, tile in enumerate(tiles):
52         parts = tile.strip().split()
53         for part in parts:
54             if part == item:
```

```

55     return int(i)
56 return -1
57

```

12.20.1.3 get_item_relative_pos()

```

def zappy.get_item_relative_pos (
    item_position )
Definition at line 5 of file zappy.py.
5 def get_item_relative_pos(item_position):
6     i = item_position
7     nb_rows = 0
8     row_length = 1
9
10    while True:
11        i -= row_length
12        if i < 0:
13            break
14        row_length += 2
15        nb_rows += 1
16
17    return i + nb_rows + 1, nb_rows
18
19 # transforms inventory string into a list of pair ressource - amount

```

12.20.1.4 go_get_item()

```

def zappy.go_get_item (
    surroundings,
    item )
Definition at line 58 of file zappy.py.
58 def go_get_item(surroundings, item):
59     distance_to_item = get_closest_of_item(surroundings, item)
60     if distance_to_item == 0:
61         return "Take " + item
62     elif distance_to_item == -1:
63         return random.choice(["Forward", "Right", "Left"])
64     else:
65         return "Forward"
66

```

12.20.1.5 how_much_of_item_here()

```

def zappy.how_much_of_item_here (
    surroundings,
    item )
Definition at line 81 of file zappy.py.
81 def how_much_of_item_here(surroundings, item):
82     if not surroundings:
83         print("is_item_here: Surroundings data is empty or None.")
84         return 0
85
86     print(surroundings)
87
88     cleaned = surroundings.strip("[ ]")
89     tiles = cleaned.split(", ")
90     here = tiles[0].strip().split()
91
92     count = 0
93     for item_here in here:
94         if item_here.strip() == item:
95             count += 1
96
97     return count

```

12.20.1.6 parse_inventory()

```

def zappy.parse_inventory (
    inventory_str )
Definition at line 20 of file zappy.py.

```

```

20 def parse_inventory(inventory_str):
21     if not inventory_str:
22         print("parse_inventory: Inventory string is empty or None.")
23         return {}
24
25     cleaned = inventory_str.strip("[ ]")
26     items = cleaned.split(", ")
27
28     resource_dict = {}
29     for item in items:
30         parts = item.strip().split()
31         if len(parts) == 2:
32             resource_name = parts[0]
33             try:
34                 amount = int(parts[1])
35                 resource_dict[resource_name] = amount
36             except ValueError:
37                 print(f"Error parsing amount for resource '{resource_name}': {parts[1]} is not an integer.")
38             continue
39     return resource_dict
40
41 # tries to find the closest item in surroundings string
42 # returns distance to the item if found, otherwise -1

```

12.21 Zappy Namespace Reference

Data Structures

- class [MapRenderer](#)

Renderer de carte responsable de l'affichage de la grille. Implémente le pattern Bridge pour séparer l'interface du rendu.

- class [MapRendererObserver](#)
- class [DetailedTileRenderStrategy](#)
- class [ITileRenderStrategy](#)
- class [ModelTileRenderStrategy](#)
- class [SimpleTileRenderStrategy](#)
- class [TileRenderStrategyFactory](#)
- class [ModelManagerAdapter](#)

Adaptateur qui permet de gérer [ModelManager](#) comme un std::shared_ptr. Cette classe encapsule le singleton [ModelManager](#) pour l'utiliser avec des shared_ptr.

12.22 ZappyTypes Namespace Reference

Namespaces

- [Colors](#)

Data Structures

- struct [Vector2](#)
- struct [Vector3](#)
- struct [Color](#)
- struct [Rectangle](#)

Enumerations

- enum [KeyboardKey](#) {
 [Z_KEY_NULL](#) = 0, [Z_KEY_APOSTROPHE](#) = 39, [Z_KEY_COMMA](#) = 44, [Z_KEY_MINUS](#) = 45,
 [Z_KEY_PERIOD](#) = 46, [Z_KEY_SLASH](#) = 47, [Z_KEY_ZERO](#) = 48, [Z_KEY_ONE](#) = 49,
 [Z_KEY_TWO](#) = 50, [Z_KEY_THREE](#) = 51, [Z_KEY_FOUR](#) = 52, [Z_KEY_FIVE](#) = 53,
 [Z_KEY_SIX](#) = 54, [Z_KEY_SEVEN](#) = 55, [Z_KEY_EIGHT](#) = 56, [Z_KEY_NINE](#) = 57,
 [Z_KEY_SEMICOLON](#) = 59, [Z_KEY_EQUAL](#) = 61, [Z_KEY_A](#) = 65, [Z_KEY_B](#) = 66,
 [Z_KEY_C](#) = 67, [Z_KEY_D](#) = 68, [Z_KEY_E](#) = 69, [Z_KEY_F](#) = 70,
 [Z_KEY_G](#) = 71, [Z_KEY_H](#) = 72, [Z_KEY_I](#) = 73, [Z_KEY_J](#) = 74,
 }

```
Z_KEY_K = 75 , Z_KEY_L = 76 , Z_KEY_M = 77 , Z_KEY_N = 78 ,
Z_KEY_O = 79 , Z_KEY_P = 80 , Z_KEY_Q = 81 , Z_KEY_R = 82 ,
Z_KEY_S = 83 , Z_KEY_T = 84 , Z_KEY_U = 85 , Z_KEY_V = 86 ,
Z_KEY_W = 87 , Z_KEY_X = 88 , Z_KEY_Y = 89 , Z_KEY_Z = 90 ,
Z_KEY_ESCAPE = 256 , Z_KEY_ENTER = 257 , Z_KEY_SPACE = 32 }
```

12.22.1 Enumeration Type Documentation

12.22.1.1 KeyboardKey

```
enum ZappyTypes::KeyboardKey
```

Enumerator

Z_KEY_NULL	
Z_KEY_APOSTROPHE	
Z_KEY_COMMA	
Z_KEY_MINUS	
Z_KEY_PERIOD	
Z_KEY_SLASH	
Z_KEY_ZERO	
Z_KEY_ONE	
Z_KEY_TWO	
Z_KEY_THREE	
Z_KEY_FOUR	
Z_KEY_FIVE	
Z_KEY_SIX	
Z_KEY_SEVEN	
Z_KEY_EIGHT	
Z_KEY_NINE	
Z_KEY_SEMICOLON	
Z_KEY_EQUAL	
Z_KEY_A	
Z_KEY_B	
Z_KEY_C	
Z_KEY_D	
Z_KEY_E	
Z_KEY_F	
Z_KEY_G	
Z_KEY_H	
Z_KEY_I	
Z_KEY_J	
Z_KEY_K	
Z_KEY_L	
Z_KEY_M	
Z_KEY_N	
Z_KEY_O	
Z_KEY_P	
Z_KEY_Q	
Z_KEY_R	
Z_KEY_S	
Z_KEY_T	

Enumerator

Z_KEY_U	
Z_KEY_V	
Z_KEY_W	
Z_KEY_X	
Z_KEY_Y	
Z_KEY_Z	
Z_KEY_ESCAPE	
Z_KEY_ENTER	
Z_KEY_SPACE	

Definition at line 67 of file Common.hpp.

```

67      {
68      Z_KEY_NULL = 0,
69      Z_KEY_APOSTROPHE = 39,
70      Z_KEY_COMMAS = 44,
71      Z_KEY_MINUS = 45,
72      Z_KEY_PERIOD = 46,
73      Z_KEY_SLASH = 47,
74      Z_KEY_ZERO = 48,
75      Z_KEY_ONE = 49,
76      Z_KEY_TWO = 50,
77      Z_KEY_THREE = 51,
78      Z_KEY_FOUR = 52,
79      Z_KEY_FIVE = 53,
80      Z_KEY_SIX = 54,
81      Z_KEY_SEVEN = 55,
82      Z_KEY_EIGHT = 56,
83      Z_KEY_NINE = 57,
84      Z_KEY_SEMICOLON = 59,
85      Z_KEY_EQUAL = 61,
86      Z_KEY_A = 65,
87      Z_KEY_B = 66,
88      Z_KEY_C = 67,
89      Z_KEY_D = 68,
90      Z_KEY_E = 69,
91      Z_KEY_F = 70,
92      Z_KEY_G = 71,
93      Z_KEY_H = 72,
94      Z_KEY_I = 73,
95      Z_KEY_J = 74,
96      Z_KEY_K = 75,
97      Z_KEY_L = 76,
98      Z_KEY_M = 77,
99      Z_KEY_N = 78,
100     Z_KEY_O = 79,
101     Z_KEY_P = 80,
102     Z_KEY_Q = 81,
103     Z_KEY_R = 82,
104     Z_KEY_S = 83,
105     Z_KEY_T = 84,
106     Z_KEY_U = 85,
107     Z_KEY_V = 86,
108     Z_KEY_W = 87,
109     Z_KEY_X = 88,
110     Z_KEY_Y = 89,
111     Z_KEY_Z = 90,
112     Z_KEY_ESCAPE = 256,
113     Z_KEY_ENTER = 257,
114     Z_KEY_SPACE = 32,
115 };

```

12.23 ZappyTypes::Colors Namespace Reference

Variables

- static constexpr ZappyTypes::Color Z_LIGHTGRAY = { 200, 200, 200, 255 }
- static constexpr ZappyTypes::Color Z_GRAY = { 130, 130, 130, 255 }
- static constexpr ZappyTypes::Color Z_DARKGRAY = { 80, 80, 80, 255 }
- static constexpr ZappyTypes::Color Z_YELLOW = { 253, 249, 0, 255 }
- static constexpr ZappyTypes::Color Z_GOLD = { 255, 203, 0, 255 }
- static constexpr ZappyTypes::Color Z_ORANGE = { 255, 161, 0, 255 }

- static constexpr `ZappyTypes::Color Z_PINK = { 255, 109, 194, 255 }`
- static constexpr `ZappyTypes::Color Z_RED = { 230, 41, 55, 255 }`
- static constexpr `ZappyTypes::Color Z_MAROON = { 190, 33, 55, 255 }`
- static constexpr `ZappyTypes::Color Z_GREEN = { 0, 228, 48, 255 }`
- static constexpr `ZappyTypes::Color Z_LIME = { 0, 158, 47, 255 }`
- static constexpr `ZappyTypes::Color Z_DARKGREEN = { 0, 117, 44, 255 }`
- static constexpr `ZappyTypes::Color Z_SKYBLUE = { 102, 191, 255, 255 }`
- static constexpr `ZappyTypes::Color Z_BLUE = { 0, 121, 241, 255 }`
- static constexpr `ZappyTypes::Color Z_DARKBLUE = { 0, 82, 172, 255 }`
- static constexpr `ZappyTypes::Color Z_PURPLE = { 200, 122, 255, 255 }`
- static constexpr `ZappyTypes::Color Z_VIOLET = { 135, 60, 190, 255 }`
- static constexpr `ZappyTypes::Color Z_DARKPURPLE = { 112, 31, 126, 255 }`
- static constexpr `ZappyTypes::Color Z_BEIGE = { 211, 176, 131, 255 }`
- static constexpr `ZappyTypes::Color Z_BROWN = { 127, 106, 79, 255 }`
- static constexpr `ZappyTypes::Color Z_DARKBROWN = { 76, 63, 47, 255 }`
- static constexpr `ZappyTypes::Color Z_WHITE = { 255, 255, 255, 255 }`
- static constexpr `ZappyTypes::Color Z_BLACK = { 0, 0, 0, 255 }`
- static constexpr `ZappyTypes::Color Z_BLANK = { 0, 0, 0, 0 }`
- static constexpr `ZappyTypes::Color Z_MAGENTA = { 255, 0, 255, 255 }`
- static constexpr `ZappyTypes::Color Z_RAYWHITE = { 245, 245, 245, 255 }`

12.23.1 Variable Documentation

12.23.1.1 Z_BEIGE

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_BEIGE = { 211, 176, 131, 255 } [inline],  
[static], [constexpr]  
Definition at line 56 of file Common.hpp.
```

12.23.1.2 Z_BLACK

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_BLACK = { 0, 0, 0, 255 } [inline], [static],  
[constexpr]  
Definition at line 60 of file Common.hpp.
```

12.23.1.3 Z_BLANK

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_BLANK = { 0, 0, 0, 0 } [inline], [static],  
[constexpr]  
Definition at line 61 of file Common.hpp.
```

12.23.1.4 Z_BLUE

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_BLUE = { 0, 121, 241, 255 } [inline], [static],  
[constexpr]  
Definition at line 51 of file Common.hpp.
```

12.23.1.5 Z_BROWN

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_BROWN = { 127, 106, 79, 255 } [inline],  
[static], [constexpr]  
Definition at line 57 of file Common.hpp.
```

12.23.1.6 Z_DARKBLUE

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_DARKBLUE = { 0, 82, 172, 255 } [inline],  
[static], [constexpr]
```

Definition at line 52 of file Common.hpp.

12.23.1.7 Z_DARKBROWN

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_DARKBROWN = { 76, 63, 47, 255 } [inline],  
[static], [constexpr]
```

Definition at line 58 of file Common.hpp.

12.23.1.8 Z_DARKGRAY

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_DARKGRAY = { 80, 80, 80, 255 } [inline],  
[static], [constexpr]
```

Definition at line 40 of file Common.hpp.

12.23.1.9 Z_DARKGREEN

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_DARKGREEN = { 0, 117, 44, 255 } [inline],  
[static], [constexpr]
```

Definition at line 49 of file Common.hpp.

12.23.1.10 Z_DARKPURPLE

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_DARKPURPLE = { 112, 31, 126, 255 } [inline],  
[static], [constexpr]
```

Definition at line 55 of file Common.hpp.

12.23.1.11 Z_GOLD

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_GOLD = { 255, 203, 0, 255 } [inline], [static],  
[constexpr]
```

Definition at line 42 of file Common.hpp.

12.23.1.12 Z_GRAY

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_GRAY = { 130, 130, 130, 255 } [inline],  
[static], [constexpr]
```

Definition at line 39 of file Common.hpp.

12.23.1.13 Z_GREEN

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_GREEN = { 0, 228, 48, 255 } [inline], [static],  
[constexpr]
```

Definition at line 47 of file Common.hpp.

12.23.1.14 Z_LIGHTGRAY

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_LIGHTGRAY = { 200, 200, 200, 255 } [inline],  
[static], [constexpr]
```

Definition at line 38 of file Common.hpp.

12.23.1.15 Z_LIME

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_LIME = { 0, 158, 47, 255 } [inline], [static],  
[constexpr]
```

Definition at line 48 of file Common.hpp.

12.23.1.16 Z_MAGENTA

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_MAGENTA = { 255, 0, 255, 255 } [inline],  
[static], [constexpr]
```

Definition at line 62 of file Common.hpp.

12.23.1.17 Z_MAROON

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_MAROON = { 190, 33, 55, 255 } [inline],  
[static], [constexpr]
```

Definition at line 46 of file Common.hpp.

12.23.1.18 Z_ORANGE

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_ORANGE = { 255, 161, 0, 255 } [inline],  
[static], [constexpr]
```

Definition at line 43 of file Common.hpp.

12.23.1.19 Z_PINK

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_PINK = { 255, 109, 194, 255 } [inline],  
[static], [constexpr]
```

Definition at line 44 of file Common.hpp.

12.23.1.20 Z_PURPLE

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_PURPLE = { 200, 122, 255, 255 } [inline],  
[static], [constexpr]
```

Definition at line 53 of file Common.hpp.

12.23.1.21 Z_RAYWHITE

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_RAYWHITE = { 245, 245, 245, 255 } [inline],  
[static], [constexpr]
```

Definition at line 63 of file Common.hpp.

12.23.1.22 Z_RED

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_RED = { 230, 41, 55, 255 } [inline], [static],  
[constexpr]
```

Definition at line 45 of file Common.hpp.

12.23.1.23 Z_SKYBLUE

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_SKYBLUE = { 102, 191, 255, 255 } [inline],  
[static], [constexpr]
```

Definition at line 50 of file Common.hpp.

12.23.1.24 Z_VIOLET

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_VIOLET = { 135, 60, 190, 255 } [inline],  
[static], [constexpr]
```

Definition at line 54 of file Common.hpp.

12.23.1.25 Z_WHITE

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_WHITE = { 255, 255, 255, 255 } [inline],  
[static], [constexpr]
```

Definition at line 59 of file Common.hpp.

12.23.1.26 Z_YELLOW

```
constexpr ZappyTypes::Color ZappyTypes::Colors::Z_YELLOW = { 253, 249, 0, 255 } [inline],  
[static], [constexpr]
```

Definition at line 41 of file Common.hpp.

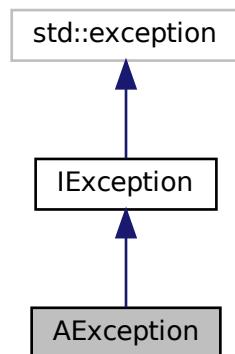
Chapter 13

Data Structure Documentation

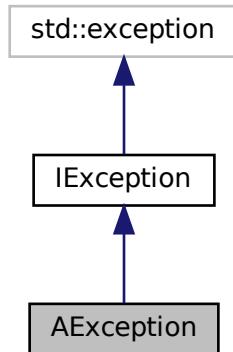
13.1 AException Class Reference

```
#include <AException.hpp>
```

Inheritance diagram for AException:



Collaboration diagram for AException:



Public Member Functions

- `AException (const std::string &type, const std::string &message)`
- virtual `~AException () noexcept=default`
- `const char * what () const noexcept override`
- `std::string getType () const noexcept override`
- `std::string getMessage () const noexcept override`
- `std::string getFormattedMessage () const noexcept override`

13.1.1 Detailed Description

Definition at line 14 of file AException.hpp.

13.1.2 Constructor & Destructor Documentation

13.1.2.1 AException()

```
AException::AException (
    const std::string & type,
    const std::string & message )
```

Definition at line 11 of file AException.cpp.

```
12     : _message(message), _type(type) {
13 }
```

13.1.2.2 ~AException()

```
virtual AException::~AException ( ) [virtual], [default], [noexcept]
```

13.1.3 Member Function Documentation

13.1.3.1 getFormattedMessage()

```
std::string AException::getFormattedMessage () const [override], [virtual], [noexcept]
Implements IException.
```

Definition at line 23 of file AException.cpp.

```
23
24     return "[" + _type + "] " + _message;
25 }
```

13.1.3.2 getMessage()

```
std::string AException::getMessage () const [override], [virtual], [noexcept]
Implements IException.
```

Definition at line 15 of file AException.cpp.

```
15
16     return _message;
17 }
```

13.1.3.3 getType()

```
std::string AException::getType () const [override], [virtual], [noexcept]
Implements IException.
```

Definition at line 19 of file AException.cpp.

```
19
20     return _type;
21 }
```

13.1.3.4 what()

```
const char * AException::what () const [override], [virtual], [noexcept]
Implements IException.
```

Definition at line 27 of file AException.cpp.

```
27
28     return getFormattedMessage().c_str();
29 }
```

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

- /root/Desktop/Zappy/src/GUI/shared/exception/AException.hpp
- /root/Desktop/Zappy/src/GUI/shared/exception/AException.cpp

13.2 agent.Agent Class Reference

Public Member Functions

- def [__init__](#) (self, ip, port, team, agent_id=0)
- def [start](#) (self)
- def [stop](#) (self)
- def [run](#) (self)
- def [send_command](#) (self, command, timeout=2.0)
- def [get_message](#) (self, timeout=None)
- def [has_messages](#) (self)
- def [process_server_message](#) (self)

Data Fields

- [ip](#)
- [port](#)
- [level](#)
- [team](#)

- [id](#)
- [map_size_x](#)
- [map_size_y](#)
- [current_behaviour](#)
- [sock](#)
- [logger](#)
- [decisionManager](#)
- [broadcastManager](#)
- [socketManager](#)

13.2.1 Detailed Description

Definition at line 11 of file agent.py.

13.2.2 Constructor & Destructor Documentation

13.2.2.1 `__init__()`

```
def agent.Agent.__init__ (
    self,
    ip,
    port,
    team,
    agent_id = 0 )
```

Definition at line 12 of file agent.py.

```
12  def __init__(self, ip, port, team, agent_id=0):
13      try:
14          self.ip = ip
15          self.port = port
16
17          self.level = 1
18          self.team = team
19          self.id = agent_id
20          self.map_size_x = None
21          self.map_size_y = None
22          self.current_behaviour = "Dyson"
23          encryption.secret_key = encryption.secret_key + self.team # to test encryption between our teams
24
25          self.sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
26          self.sock.connect((self.ip, self.port))
27
28          self.logger = Logger("AI.log", message_prefix=f"(Agent nř{self.id}): ")
29
30          self.decisionManager = DecisionManager(self)
31          self.broadcastManager = BroadcastManager(self)
32          self.socketManager = SocketManager(self.sock)
33          self.socketManager.start()
34
35      except socket.error as e:
36          print(f"Error connecting to server: {e}")
37          sys.exit(1)
38      except Exception as e:
39          print(f"Unexpected error: {e}")
40          sys.exit(1)
41
42  finally:
43      print(f"Connected to {self.ip}:{self.port} as team '{self.team}' with agent ID {self.id}.")
```

13.2.3 Member Function Documentation

13.2.3.1 get_message()

```
def agent.Agent.get_message (
    self,
    timeout = None )
```

Definition at line 98 of file agent.py.

```
98     def get_message(self, timeout=None):
99         return self.socketManager.get_message(timeout=timeout)
100
```

13.2.3.2 has_messages()

```
def agent.Agent.has_messages (
    self )
```

Definition at line 101 of file agent.py.

```
101    def has_messages(self):
102        return self.socketManager.has_messages()
103
```

13.2.3.3 process_server_message()

```
def agent.Agent.process_server_message (
    self )
```

Definition at line 104 of file agent.py.

```
104    def process_server_message(self):
105        if not self.has_messages():
106            return
107        message = self.get_message()
108        if message.startswith("message "):
109            self.broadcastManager.manage_broadcast(message)
110        elif message.startswith("dead"):
111            print("Agent has died.")
112            self.stop()
113        elif message.startswith("Current level: "):
114            try:
115                self.level = int(message.split(": ")[1])
116                if self.level >= 2:
117                    self.current_behaviour = "GetFoodAndMinerals"
118                    print(f"Current level set to: {self.level}")
119                except ValueError:
120                    print(f"Failed to parse level from message: {message}")
121            else:
122                print(f"Unknown server message: {message}")
```

13.2.3.4 run()

```
def agent.Agent.run (
    self )
```

Definition at line 78 of file agent.py.

```
78    def run(self):
79        while self.socketManager.running:
80            try:
81
82                self.process_server_message()
83                self.decisionManager.take_action()
84                sleep(0.1)
85
86            except BrokenPipeError:
87                print(f"Agent {self.id}: Connection closed by server.")
88                self.stop()
89                break
90            except Exception as e:
91                print(f"Agent {self.id}: Error: {e}")
92                self.stop()
93                break
94
```

13.2.3.5 send_command()

```
def agent.Agent.send_command (
```

```

        self,
        command,
        timeout = 2.0 )

```

Definition at line 95 of file agent.py.

```

95  def send_command(self, command, timeout=2.0):
96      return self.socketManager.send_command(command, timeout=timeout)
97

```

13.2.3.6 start()

```

def agent.Agent.start (
    self )

```

Definition at line 46 of file agent.py.

```

46  def start(self):
47      welcome_msg = self.get_message(timeout=4)
48      team_slots = self.send_command(self.team)
49      map_size = self.get_message(timeout=4)
50
51      self.map_size_x = int(map_size.split()[0])
52      self.map_size_y = int(map_size.split()[1])
53
54      if welcome_msg == "WELCOME":
55          print(f"Welcome message {welcome_msg}")
56      else:
57          print(f"Unexpected welcome message: {welcome_msg}")
58          sys.exit(1)
59
60      if (team_slots == "ko"):
61          print("Failed to join team. Either the team is full, or its name is incorrect.")
62          sys.exit(1)
63      else:
64          print(f"Joined team {self.team} successfully, {team_slots} slots left in the team.")
65
66      print(f"Map size: {map_size}")
67
68      self.run()
69
70

```

13.2.3.7 stop()

```

def agent.Agent.stop (
    self )

```

Definition at line 71 of file agent.py.

```

71  def stop(self):
72      self.logger.info(f"Stopping agent {self.id}...")
73      self.socketManager.stop()
74      self.sock.close()
75      print(f"Agent {self.id} stopped.")
76
77

```

13.2.4 Field Documentation

13.2.4.1 broadcastManager

agent.Agent.broadcastManager

Definition at line 31 of file agent.py.

13.2.4.2 current_behaviour

agent.Agent.current_behaviour

Definition at line 22 of file agent.py.

13.2.4.3 decisionManager

agent.Agent.decisionManager
Definition at line 30 of file agent.py.

13.2.4.4 id

agent.Agent.id
Definition at line 19 of file agent.py.

13.2.4.5 ip

agent.Agent.ip
Definition at line 14 of file agent.py.

13.2.4.6 level

agent.Agent.level
Definition at line 17 of file agent.py.

13.2.4.7 logger

agent.Agent.logger
Definition at line 28 of file agent.py.

13.2.4.8 map_size_x

agent.Agent.map_size_x
Definition at line 20 of file agent.py.

13.2.4.9 map_size_y

agent.Agent.map_size_y
Definition at line 21 of file agent.py.

13.2.4.10 port

agent.Agent.port
Definition at line 15 of file agent.py.

13.2.4.11 sock

agent.Agent.sock
Definition at line 25 of file agent.py.

13.2.4.12 socketManager

agent.Agent.socketManager
Definition at line 32 of file agent.py.

13.2.4.13 team

agent.Agent.team

Definition at line 18 of file agent.py.

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

- /root/Desktop/Zappy/src/AI/agent/[agent.py](#)

13.3 raylibcpp::Audio Class Reference

```
#include <Audio.hpp>
```

Public Member Functions

- [Audio \(\)](#)
- [~Audio \(\)](#)

13.3.1 Detailed Description

Definition at line 14 of file Audio.hpp.

13.3.2 Constructor & Destructor Documentation

13.3.2.1 [Audio\(\)](#)

```
raylibcpp::Audio::Audio ()
```

Definition at line 14 of file Audio.cpp.

```
14     {
15     InitAudioDevice();
16 }
```

13.3.2.2 [~Audio\(\)](#)

```
raylibcpp::Audio::~Audio ()
```

Definition at line 18 of file Audio.cpp.

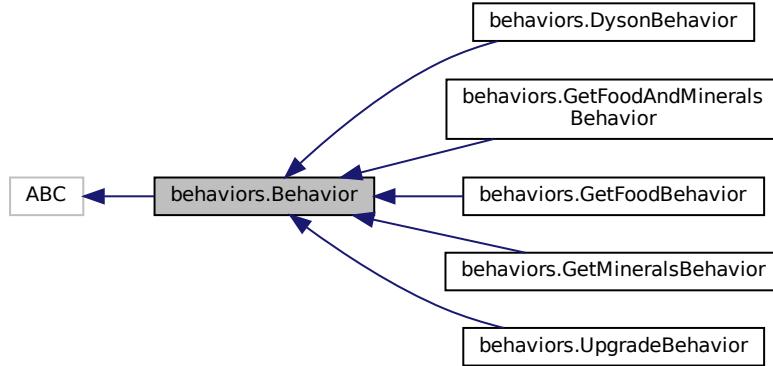
```
18 {
19     CloseAudioDevice();
20 }
```

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

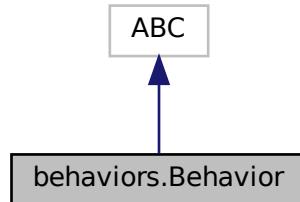
- /root/Desktop/Zappy/libs/RaylibCPP/audio/[Audio.hpp](#)
- /root/Desktop/Zappy/libs/RaylibCPP/audio/[Audio.cpp](#)

13.4 behaviors.Behavior Class Reference

Inheritance diagram for behaviors.Behavior:



Collaboration diagram for behaviors.Behavior:



Public Member Functions

- def `__init__` (self, agent)
- def `execute` (self, surroundings=None, inventory=None)

Data Fields

- `agent`

13.4.1 Detailed Description

Definition at line 8 of file behaviors.py.

13.4.2 Constructor & Destructor Documentation

13.4.2.1 __init__()

```
def behaviors.Behavior.__init__ (
    self,
    agent )
Definition at line 9 of file behaviors.py.
9  def __init__(self, agent):
10     self.agent = agent
```

13.4.3 Member Function Documentation

13.4.3.1 execute()

```
def behaviors.Behavior.execute (
    self,
    surroundings = None,
    inventory = None )
```

Reimplemented in [behaviors.GetFoodAndMineralsBehavior](#), [behaviors.DysonBehavior](#), [behaviors.GetMineralsBehavior](#), [behaviors.UpgradeBehavior](#), and [behaviors.GetFoodBehavior](#).

Definition at line 12 of file behaviors.py.

```
12  def execute(self, surroundings=None, inventory=None):
13      pass
14
15
```

13.4.4 Field Documentation

13.4.4.1 agent

`behaviors.Behavior.agent`

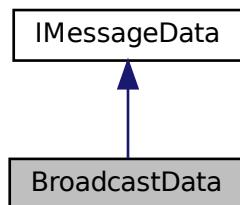
Definition at line 10 of file behaviors.py.

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

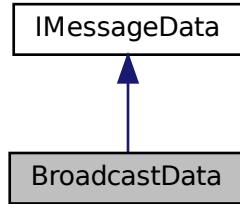
- /root/Desktop/Zappy/src/AI/agent/[behaviors.py](#)

13.5 BroadcastData Class Reference

```
#include <BroadcastData.hpp>
Inheritance diagram for BroadcastData:
```



Collaboration diagram for BroadcastData:



Public Member Functions

- `BroadcastData (int playerId, const std::string &message)`
- `MessageType getType () const override`
- `int getPlayerId () const`
- `const std::string & getMessage () const`
- `void setPlayerId (int value)`
- `void setMessage (const std::string &value)`

13.5.1 Detailed Description

Definition at line 14 of file BroadcastData.hpp.

13.5.2 Constructor & Destructor Documentation

13.5.2.1 BroadcastData()

```
BroadcastData::BroadcastData (
    int playerId,
    const std::string & message ) [inline]
Definition at line 16 of file BroadcastData.hpp.
17 : _playerId(playerId), _message(message) {}
```

13.5.3 Member Function Documentation

13.5.3.1 getMessage()

```
const std::string& BroadcastData::getMessage () const [inline]
Definition at line 21 of file BroadcastData.hpp.
21 { return _message; }
```

13.5.3.2 getPlayerId()

```
int BroadcastData::getPlayerId () const [inline]
Definition at line 20 of file BroadcastData.hpp.
20 { return _playerId; }
```

13.5.3.3 getType()

```
MessageType BroadcastData::getType () const [inline], [override], [virtual]
Implements IMessageData.
Definition at line 18 of file BroadcastData.hpp.
18 { return MessageType::Broadcast; }
```

13.5.3.4 setMessage()

```
void BroadcastData::setMessage (
    const std::string & value ) [inline]
Definition at line 24 of file BroadcastData.hpp.
24 { _message = value; }
```

13.5.3.5 setPlayerId()

```
void BroadcastData::setPlayerId (
    int value ) [inline]
Definition at line 23 of file BroadcastData.hpp.
23 { _playerId = value; }
```

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

- /root/Desktop/Zappy/src/GUI/network/protocol/messageData/[BroadcastData.hpp](#)

13.6 broadcastManager.BroadcastManager Class Reference

Public Member Functions

- def [__init__](#) (self, agent)
- def [manage_broadcast](#) (self, full_server_message)

Data Fields

- [agent](#)

13.6.1 Detailed Description

Definition at line 5 of file broadcastManager.py.

13.6.2 Constructor & Destructor Documentation

13.6.2.1 __init__()

```
def broadcastManager.BroadcastManager.__init__ (
    self,
    agent )
Definition at line 6 of file broadcastManager.py.
6  def __init__(self, agent):
7      self.agent = agent
8
```

13.6.3 Member Function Documentation

13.6.3.1 manage_broadcast()

```
def broadcastManager.BroadcastManager.manage_broadcast (
    self,
    full_server_message )
Definition at line 9 of file broadcastManager.py.
9  def manage_broadcast(self, full_server_message):
10     if not full_server_message or len(full_server_message) < 2:
11         print("Empty broadcast message received.")
12         return
13     if not full_server_message.startswith("message "):
14         print(f"Invalid broadcast message format: {full_server_message}")
15         return
16
17     try:
18         emitter_direction = int(full_server_message.split(" ")[1].split(",")[0])
19     except ValueError:
20         print(f"Invalid emitter direction in broadcast message: {full_server_message}")
21         return
22
23     broadcast_message = full_server_message.split(", ", [1]
24
25     if not broadcast_message:
26         print("Empty broadcast message content.")
27         return
28
29     print(f"Broadcast received from direction: {emitter_direction}")
30
31     decrypted_message = encryption.decrypt_message(full_server_message.split(", ", [1])
32
33     if decrypted_message is not None:
34         print(f"Decrypted broadcast message: {decrypted_message}")
35     else:
36         print(f"Enemy broadcast message: {broadcast_message}")
37         return
38
39     if decrypted_message.startswith("HELP! Upgrade: "):
40         try:
41             level = int(decrypted_message.split(": ")[1])
42             if level == self.agent.level + 1:
43                 actions.got_to_dir(self.agent, emitter_direction)
44                 print(f"Received help request for level {level}. Moving towards the emitter.")
45             else:
46                 print(f"Received help request for level {level}, but current level is {self.agent.level}.
47 Ignoring.")
47         except ValueError:
48             print(f"Failed to parse level from message: {decrypted_message}")
```

13.6.4 Field Documentation

13.6.4.1 agent

broadcastManager.BroadcastManager.agent

Definition at line 7 of file broadcastManager.py.

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

- /root/Desktop/Zappy/src/AI/agent/[broadcastManager.py](#)

13.7 RayGUICPP::Button Class Reference

```
#include <RayGuiEncap.hpp>
```

Static Public Member Functions

- static bool [Draw](#) (const Rectangle &bounds, const std::string &text)

13.7.1 Detailed Description

Definition at line 34 of file RayGuiEncap.hpp.

13.7.2 Member Function Documentation

13.7.2.1 Draw()

```
static bool RayGUICPP::Button::Draw (
    const Rectangle & bounds,
    const std::string & text) [inline], [static]
Definition at line 36 of file RayGuiEncap.hpp.
36
37     return GuiButton(bounds, text.c_str());
38 }
```

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

- /root/Desktop/Zappy/libs/RayGUICPP/include/RayGuiEncap.hpp

13.8 raylibcpp::Camera2DWrap Class Reference

```
#include <Camera.hpp>
```

Public Member Functions

- [Camera2DWrap \(\)](#)
- [~Camera2DWrap \(\)](#)
- [Camera2D & get \(\)](#)

13.8.1 Detailed Description

Definition at line 13 of file Camera.hpp.

13.8.2 Constructor & Destructor Documentation

13.8.2.1 Camera2DWrap()

```
raylibcpp::Camera2DWrap::Camera2DWrap( )
Definition at line 12 of file Camera.cpp.
12
13     camera.offset = (Vector2){0.0f, 0.0f};
14     camera.target = (Vector2){0.0f, 0.0f};
15     camera.rotation = 0.0f;
16     camera.zoom = 1.0f;
17 }
```

13.8.2.2 ~Camera2DWrap()

```
raylibcpp::Camera2DWrap::~Camera2DWrap( )
Definition at line 19 of file Camera.cpp.
19 {}
```

13.8.3 Member Function Documentation

13.8.3.1 get()

```
Camera2D & raylibcpp::Camera2DWrap::get ( )
```

Definition at line 21 of file Camera.cpp.

```
21
22     return camera;
23 }
```

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

- /root/Desktop/Zappy/libs/RaylibCPP/camera/[Camera.hpp](#)
- /root/Desktop/Zappy/libs/RaylibCPP/camera/[Camera.cpp](#)

13.9 raylibcpp::Camera3DWrap Class Reference

```
#include <Camera.hpp>
```

Public Member Functions

- [Camera3DWrap \(\)](#)
- [Camera3DWrap \(Vector3 position, Vector3 target, Vector3 up, float fovy, int projection=CAMERA_PERSPECTIVE\)](#)
- [~Camera3DWrap \(\)](#)
- Camera3D & [get \(\)](#)
- void [beginMode \(\)](#)
- void [endMode \(\)](#)

13.9.1 Detailed Description

Definition at line 22 of file Camera.hpp.

13.9.2 Constructor & Destructor Documentation

13.9.2.1 Camera3DWrap() [1/2]

```
raylibcpp::Camera3DWrap::Camera3DWrap ( )
```

Definition at line 25 of file Camera.cpp.

```
25
26     camera.position = (Vector3){4.0f, 4.0f, 10.0f};
27     camera.target = (Vector3){0.0f, 1.0f, 0.0f};
28     camera.up = (Vector3){0.0f, 1.0f, 0.0f};
29     camera.fovy = 45.0f;
30     camera.projection = CAMERA_PERSPECTIVE;
31 }
```

13.9.2.2 Camera3DWrap() [2/2]

```
raylibcpp::Camera3DWrap::Camera3DWrap (
    Vector3 position,
    Vector3 target,
    Vector3 up,
    float fovy,
    int projection = CAMERA_PERSPECTIVE )
```

Definition at line 33 of file Camera.cpp.

```
33
34     camera.position = position;
35     camera.target = target;
36     camera.up = up;
37     camera.fovy = fovy;
38     camera.projection = projection;
39 }
```

13.9.2.3 ~Camera3DWrap()

```
raylibcpp::Camera3DWrap::~Camera3DWrap ( )
Definition at line 41 of file Camera.cpp.
41 { }
```

13.9.3 Member Function Documentation

13.9.3.1 beginMode()

```
void raylibcpp::Camera3DWrap::beginMode ( )
Definition at line 47 of file Camera.cpp.
47 {
48     BeginMode3D(camera);
49 }
```

13.9.3.2 endMode()

```
void raylibcpp::Camera3DWrap::endMode ( )
Definition at line 51 of file Camera.cpp.
51 {
52     EndMode3D();
53 }
```

13.9.3.3 get()

```
Camera3D & raylibcpp::Camera3DWrap::get ( )
Definition at line 43 of file Camera.cpp.
43 {
44     return camera;
45 }
```

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

- /root/Desktop/Zappy/libs/RaylibCPP/camera/[Camera.hpp](#)
- /root/Desktop/Zappy/libs/RaylibCPP/camera/[Camera.cpp](#)

13.10 CameraController Class Reference

```
#include <CameraController.hpp>
```

Public Member Functions

- [CameraController](#) (float initialDistance=25.0f)
- [~CameraController](#) ()=default
- void [init](#) (std::shared_ptr<[IGraphicsLib](#)> graphics)
- void [update](#) (std::shared_ptr<[IGraphicsLib](#)> graphics)
- void [reset](#) ()
- void [handleMouseInput](#) (std::shared_ptr<[IGraphicsLib](#)> graphics)
- void [handleKeyboardInput](#) (std::shared_ptr<[IGraphicsLib](#)> graphics)
- void [updateCameraPosition](#) (std::shared_ptr<[IGraphicsLib](#)> graphics)
- bool [isDragging](#) () const
- void [setDragging](#) (bool dragging)
- void [setMousePosition](#) (int x, int y)
- [ZappyTypes::Vector3](#) & [position](#) ()
- [ZappyTypes::Vector3](#) & [target](#) ()
- [ZappyTypes::Vector3](#) & [up](#) ()
- float & [distance](#) ()

- float & `angleX()`
- float & `angleY()`
- void `setMapDimensions(int width, int height)`

13.10.1 Detailed Description

Definition at line 14 of file CameraController.hpp.

13.10.2 Constructor & Destructor Documentation

13.10.2.1 CameraController()

```
CameraController::CameraController (
    float initialDistance = 25.0f ) [explicit]
```

Definition at line 15 of file CameraController.cpp.

```
16     : m_cameraPosition({15.0f, 15.0f, 15.0f}),
17     m_cameraTarget({0.0f, 0.0f, 0.0f}),
18     m_cameraUp({0.0f, 1.0f, 0.0f}),
19     m_initialCameraTarget({0.0f, 0.0f, 0.0f}),
20     m_isDragging(false),
21     m_lastMouseX(0),
22     m_lastMouseY(0),
23     m_cameraDistance(initialDistance),
24     m_cameraAngleY(0.8f),
25     m_cameraAngleX(0.6f),
26     m_mouseSensitivity(0.003f),
27     m_keyboardSpeed(0.1f) {
28 }
```

13.10.2.2 ~CameraController()

```
CameraController::~CameraController () [default]
```

13.10.3 Member Function Documentation

13.10.3.1 angleX()

```
float& CameraController::angleX () [inline]
```

Definition at line 36 of file CameraController.hpp.

```
36 { return m_cameraAngleX; }
```

13.10.3.2 angleY()

```
float& CameraController::angleY () [inline]
```

Definition at line 37 of file CameraController.hpp.

```
37 { return m_cameraAngleY; }
```

13.10.3.3 distance()

```
float& CameraController::distance () [inline]
```

Definition at line 35 of file CameraController.hpp.

```
35 { return m_cameraDistance; }
```

13.10.3.4 handleKeyboardInput()

```
void CameraController::handleKeyboardInput (
    std::shared_ptr< IGraphicsLib > graphics )
Definition at line 94 of file CameraController.cpp.
94
95     if (graphics->IsKeyPressed(KEY_R)) {
96         reset();
97     }
98     ZappyTypes::Vector3 forward = {
99         std::cos(m_cameraAngleY),
100        0.0f,
101        std::sin(m_cameraAngleY)
102    };
103    float forwardLength = std::sqrt(forward.x * forward.x + forward.z * forward.z);
104    if (forwardLength > 0.0001f) {
105        forward.x /= forwardLength;
106        forward.z /= forwardLength;
107    }
108    ZappyTypes::Vector3 right = {
109        -forward.z,
110        0.0f,
111        forward.x
112    };
113
114    ZappyTypes::Vector3 newTarget = m_cameraTarget;
115    if (graphics->IsKeyDown(KEY_S) || graphics->IsKeyDown(KEY_DOWN)) {
116        newTarget.x += forward.x * m_keyboardSpeed;
117        newTarget.z += forward.z * m_keyboardSpeed;
118    }
119    if (graphics->IsKeyDown(KEY_W) || graphics->IsKeyDown(KEY_Z) || graphics->IsKeyDown(KEY_UP)) {
120        newTarget.x -= forward.x * m_keyboardSpeed;
121        newTarget.z -= forward.z * m_keyboardSpeed;
122    }
123    if (graphics->IsKeyDown(KEY_D) || graphics->IsKeyDown(KEY_RIGHT)) {
124        newTarget.x -= right.x * m_keyboardSpeed;
125        newTarget.z -= right.z * m_keyboardSpeed;
126    }
127    if (graphics->IsKeyDown(KEY_Q) || graphics->IsKeyDown(KEY_A) || graphics->IsKeyDown(KEY_LEFT)) {
128        newTarget.x += right.x * m_keyboardSpeed;
129        newTarget.z += right.z * m_keyboardSpeed;
130    }
131    float halfWidth = m_mapWidth * 1.2f;
132    float halfHeight = m_mapHeight * 1.2f;
133    float margin = 2.0f;
134    newTarget.x = std::max(std::min(newTarget.x, halfWidth - margin), -halfWidth + margin);
135    newTarget.z = std::max(std::min(newTarget.z, halfHeight - margin), -halfHeight + margin);
136    m_cameraTarget = newTarget;
137 }
```

13.10.3.5 handleMouseInput()

```
void CameraController::handleMouseInput (
    std::shared_ptr< IGraphicsLib > graphics )
```

Definition at line 52 of file CameraController.cpp.

```
52
53     if (!graphics) {
54         return;
55     }
56     if (graphics->IsMouseButtonPressed(0)) {
57         m_isDragging = true;
58         m_lastMouseX = graphics->GetMouseX();
59         m_lastMouseY = graphics->GetMouseY();
60     }
61
62     if (graphics->IsMouseButtonReleased(0)) {
63         m_isDragging = false;
64     }
65
66     if (m_isDragging) {
67         int currentMouseX = graphics->GetMouseX();
68         int currentMouseY = graphics->GetMouseY();
69
70         float deltaX = static_cast<float>(currentMouseX - m_lastMouseX);
71         float deltaY = static_cast<float>(currentMouseY - m_lastMouseY);
72
73         m_cameraAngleY += deltaX * m_mouseSensitivity;
74         m_cameraAngleX -= deltaY * m_mouseSensitivity;
75
76         if (m_cameraAngleX > 1.5f) m_cameraAngleX = 1.5f;
77         if (m_cameraAngleX < -1.5f) m_cameraAngleX = -1.5f;
78 }
```

```

79     m_lastMouseX = currentMouseX;
80     m_lastMouseY = currentMouseY;
81 }
82
83 float wheelMovement = graphics->GetMouseWheelMove();
84 if (wheelMovement != 0) {
85     float zoomSpeed = m_cameraDistance * 0.1f;
86     if (zoomSpeed < 0.5f) zoomSpeed = 0.5f;
87     if (zoomSpeed > 3.0f) zoomSpeed = 3.0f;
88     m_cameraDistance -= wheelMovement * zoomSpeed;
89     if (m_cameraDistance < 2.0f) m_cameraDistance = 2.0f;
90     if (m_cameraDistance > 50.0f) m_cameraDistance = 50.0f;
91 }
92 }
```

13.10.3.6 init()

```
void CameraController::init (
    std::shared_ptr< IGraphicsLib > graphics )
```

Definition at line 30 of file CameraController.cpp.

```

30
31     updateCameraPosition(graphics);
```

{

13.10.3.7 isDragging()

```
bool CameraController::isDragging () const [inline]
```

Definition at line 27 of file CameraController.hpp.

```

27 { return m_isDragging; }
```

13.10.3.8 position()

```
ZappyTypes::Vector3& CameraController::position () [inline]
```

Definition at line 31 of file CameraController.hpp.

```

31 { return m_cameraPosition; }
```

13.10.3.9 reset()

```
void CameraController::reset ()
```

Definition at line 40 of file CameraController.cpp.

```

40
41     m_cameraDistance = 25.0f;
42     m_cameraAngleY = 0.8f;
43     m_cameraAngleX = 0.6f;
44     m_cameraTarget = m_initialCameraTarget;
45 }
```

13.10.3.10 setDragging()

```
void CameraController::setDragging (
    bool dragging ) [inline]
```

Definition at line 28 of file CameraController.hpp.

```

28 { m_isDragging = dragging; }
```

13.10.3.11 setMapDimensions()

```
void CameraController::setMapDimensions (
    int width,
    int height ) [inline]
```

Definition at line 40 of file CameraController.hpp.

```

40
41     m_mapWidth = width;
42     m_mapHeight = height;
```

```
43     }
```

13.10.3.12 setMousePosition()

```
void CameraController::setMousePosition (
    int x,
    int y)
Definition at line 47 of file CameraController.cpp.
```

```
47
48     m_lastMouseX = x;
49     m_lastMouseY = y;
50 }
```

13.10.3.13 target()

```
ZappyTypes::Vector3& CameraController::target () [inline]
Definition at line 32 of file CameraController.hpp.
32 { return m_cameraTarget; }
```

13.10.3.14 up()

```
ZappyTypes::Vector3& CameraController::up () [inline]
Definition at line 33 of file CameraController.hpp.
33 { return m_cameraUp; }
```

13.10.3.15 update()

```
void CameraController::update (
    std::shared_ptr< IGraphicsLib > graphics )
Definition at line 34 of file CameraController.cpp.
```

```
34
35     handleMouseInput(graphics);
36     handleKeyboardInput(graphics);
37     updateCameraPosition(graphics);
38 }
```

13.10.3.16 updateCameraPosition()

```
void CameraController::updateCameraPosition (
    std::shared_ptr< IGraphicsLib > graphics )
Definition at line 139 of file CameraController.cpp.
```

```
139
140     if (m_cameraDistance < 2.0f) m_cameraDistance = 2.0f;
141     if (m_cameraDistance > 50.0f) m_cameraDistance = 50.0f;
142     if (m_cameraAngleX < 0.1f) m_cameraAngleX = 0.1f;
143     if (m_cameraAngleX > 1.5f) m_cameraAngleX = 1.5f;
144     m_cameraPosition.x = m_cameraTarget.x + m_cameraDistance * std::cos(m_cameraAngleY) *
        std::cos(m_cameraAngleX);
145     m_cameraPosition.y = m_cameraTarget.y + m_cameraDistance * std::sin(m_cameraAngleX);
146     m_cameraPosition.z = m_cameraTarget.z + m_cameraDistance * std::sin(m_cameraAngleY) *
        std::cos(m_cameraAngleX);
147     if (m_cameraPosition.y < 1.0f) {
148         m_cameraPosition.y = 1.0f;
149     }
150     graphics->CreateCamera3D(m_cameraPosition, m_cameraTarget, m_cameraUp, 45.0f, 0);
151 }
```

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

- /root/Desktop/Zappy/src/GUI/cameraController/[CameraController.hpp](#)
- /root/Desktop/Zappy/src/GUI/cameraController/[CameraController.cpp](#)

13.11 RayGUICPP::CheckBox Class Reference

```
#include <RayGuiEncap.hpp>
```

Static Public Member Functions

- static bool [Draw](#) (const Rectangle &bounds, const std::string &text, bool checked)

13.11.1 Detailed Description

Definition at line 48 of file RayGuiEncap.hpp.

13.11.2 Member Function Documentation

13.11.2.1 Draw()

```
static bool RayGUICPP::CheckBox::Draw (
    const Rectangle & bounds,
    const std::string & text,
    bool checked ) [inline], [static]
```

Definition at line 50 of file RayGuiEncap.hpp.

```
50
51     bool value = checked;
52     GuiCheckBox(bounds, text.c_str(), &value);
53     return value;
54 }
```

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

- /root/Desktop/Zappy/libs/RayGUICPP/include/[RayGuiEncap.hpp](#)

13.12 circular_buffer_s Struct Reference

```
#include <circular_buffer.h>
```

Data Fields

- char [buffer](#) [[BUFFER_SIZE](#)]
- int [start](#)
- int [end](#)
- int [count](#)

13.12.1 Detailed Description

Definition at line 12 of file circular_buffer.h.

13.12.2 Field Documentation

13.12.2.1 buffer

```
char circular_buffer_s::buffer[BUFFER\_SIZE]
```

Definition at line 13 of file circular_buffer.h.

13.12.2.2 count

```
int circular_buffer_s::count
```

Definition at line 16 of file `circular_buffer.h`.

13.12.2.3 end

```
int circular_buffer_s::end
```

Definition at line 15 of file `circular_buffer.h`.

13.12.2.4 start

```
int circular_buffer_s::start
```

Definition at line 14 of file `circular_buffer.h`.

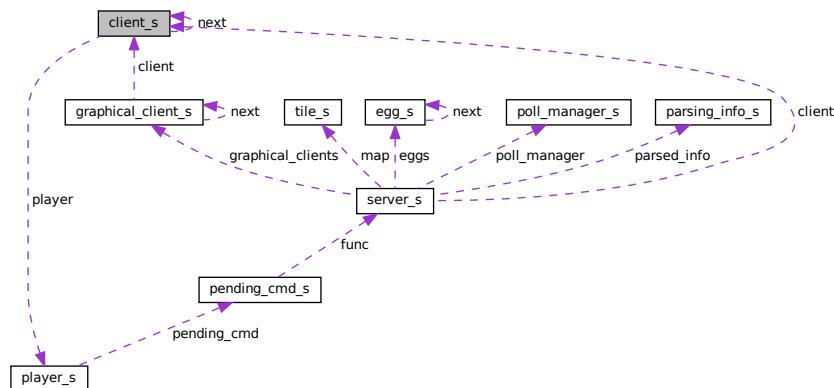
The documentation for this struct was generated from the following file:

- /root/Desktop/Zappy/src/Server/include/circular_buffer.h

13.13 client_s Struct Reference

```
#include <client.h>
```

Collaboration diagram for `client_s`:



Data Fields

- int `client_id`
- int `client_fd`
- struct `pollfd * client_poll`
- struct `sockaddr_in * client_add`
- `player_t * player`
- struct `client_s * next`
- bool `is_fully_connected`
- enum `client_type_e type`

13.13.1 Detailed Description

Definition at line 24 of file `client.h`.

13.13.2 Field Documentation

13.13.2.1 client_add

```
struct sockaddr_in* client_s::client_add
```

Definition at line 28 of file client.h.

13.13.2.2 client_fd

```
int client_s::client_fd
```

Definition at line 26 of file client.h.

13.13.2.3 client_id

```
int client_s::client_id
```

Definition at line 25 of file client.h.

13.13.2.4 client_poll

```
struct pollfd* client_s::client_poll
```

Definition at line 27 of file client.h.

13.13.2.5 is_fully_connected

```
bool client_s::is_fully_connected
```

Definition at line 31 of file client.h.

13.13.2.6 next

```
struct client_s* client_s::next
```

Definition at line 30 of file client.h.

13.13.2.7 player

```
player_t* client_s::player
```

Definition at line 29 of file client.h.

13.13.2.8 type

```
enum client_type_e client_s::type
```

Definition at line 31 of file client.h.

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

- /root/Desktop/Zappy/src/Server/include/client.h

13.14 ZappyTypes::Color Struct Reference

```
#include <Common.hpp>
```

Data Fields

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

13.14.1 Detailed Description

Definition at line 23 of file Common.hpp.

13.14.2 Field Documentation

13.14.2.1 [a](#)

```
unsigned char ZappyTypes::Color::a
```

Definition at line 27 of file Common.hpp.

13.14.2.2 [b](#)

```
unsigned char ZappyTypes::Color::b
```

Definition at line 26 of file Common.hpp.

13.14.2.3 [g](#)

```
unsigned char ZappyTypes::Color::g
```

Definition at line 25 of file Common.hpp.

13.14.2.4 [r](#)

```
unsigned char ZappyTypes::Color::r
```

Definition at line 24 of file Common.hpp.

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

- /root/Desktop/Zappy/src/Shared/[Common.hpp](#)

13.15 RayGUICPP::ColorPicker Class Reference

```
#include <RayGuiEncap.hpp>
```

Static Public Member Functions

- static Color [Draw](#) (const Rectangle &bounds, Color color)

13.15.1 Detailed Description

Definition at line 218 of file RayGuiEncap.hpp.

13.15.2 Member Function Documentation

13.15.2.1 Draw()

```
static Color RayGUICPP::ColorPicker::Draw (
    const Rectangle & bounds,
    Color color ) [inline], [static]
```

Definition at line 220 of file RayGuiEncap.hpp.

```
220
221     GuiColorPicker(bounds, nullptr, &color);
222     return color;
223 }
```

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

- /root/Desktop/Zappy/libs/RayGUICPP/include/RayGuiEncap.hpp

13.16 RayGUICPP::ComboBox Class Reference

```
#include <RayGuiEncap.hpp>
```

Static Public Member Functions

- static int [Draw](#) (const Rectangle &bounds, const std::string &text, int active)

13.16.1 Detailed Description

Definition at line 77 of file RayGuiEncap.hpp.

13.16.2 Member Function Documentation

13.16.2.1 Draw()

```
static int RayGUICPP::ComboBox::Draw (
    const Rectangle & bounds,
    const std::string & text,
    int active ) [inline], [static]
```

Definition at line 79 of file RayGuiEncap.hpp.

```
79
80     int value = active;
81     GuiComboBox(bounds, text.c_str(), &value);
82     return value;
83 }
```

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

- /root/Desktop/Zappy/libs/RayGUICPP/include/RayGuiEncap.hpp

13.17 command_data_s Struct Reference

```
#include <command.h>
```

Data Fields

- const char ** [commands](#)
- void(** [functions](#))(server_t *, client_t *, char **)
- int * [times](#)
- enum [client_type_e](#) * [accepted_types](#)

13.17.1 Detailed Description

Definition at line 15 of file command.h.

13.17.2 Field Documentation

13.17.2.1 accepted_types

```
enum client\_type\_e* command_data_s::accepted_types
Definition at line 19 of file command.h.
```

13.17.2.2 commands

```
const char** command_data_s::commands
Definition at line 16 of file command.h.
```

13.17.2.3 functions

```
void(** command_data_s::functions) (server\_t *, client\_t *, char **)
Definition at line 17 of file command.h.
```

13.17.2.4 times

```
int* command_data_s::times
Definition at line 18 of file command.h.
```

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

- /root/Desktop/Zappy/src/Server/include/[command.h](#)

13.18 decisionManager.DecisionManager Class Reference

Public Member Functions

- def [__init__](#) (self, agent)
- def [take_action](#) (self)

Data Fields

- [agent](#)
- [behaviors](#)

13.18.1 Detailed Description

Definition at line 7 of file decisionManager.py.

13.18.2 Constructor & Destructor Documentation

13.18.2.1 [__init__\(\)](#)

```
def decisionManager.DecisionManager.__init__ (
    self,
    agent )
Definition at line 8 of file decisionManager.py.
8  def __init__(self, agent):
9      self.agent = agent
10     self.behaviors = {
11         "GetFood": behaviors.GetFoodBehavior(agent),
12         "GetMinerals": behaviors.GetMineralsBehavior(agent),
```

```
13     "GetFoodAndMinerals": behaviors.GetFoodAndMineralsBehavior(agent),
14     "Upgrade": behaviors.UpgradeBehavior(agent),
15     "Dyson": behaviors.DysonBehavior(agent)
16   }
17 
```

13.18.3 Member Function Documentation

13.18.3.1 take_action()

```
def decisionManager.DecisionManager.take_action (
    self )
Definition at line 18 of file decisionManager.py.
18 def take_action(self):
19     inventory = self.agent.send_command("Inventory")
20     surroundings = self.agent.send_command("Look")
21
22     if inventory is None or surroundings is None:
23         print("Failed to retrieve inventory or surroundings.")
24         return
25
26     print(inventory)
27
28     self.behaviors[self.agent.current_behaviour].execute(surroundings, inventory)
29     self.behaviors["Upgrade"].execute(surroundings, inventory)
```

13.18.4 Field Documentation

13.18.4.1 agent

```
decisionManager.DecisionManager.agent
Definition at line 9 of file decisionManager.py.
```

13.18.4.2 behaviors

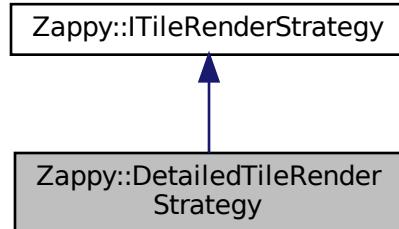
```
decisionManager.DecisionManager.behaviors
Definition at line 10 of file decisionManager.py.
The documentation for this class was generated from the following file:
```

- /root/Desktop/Zappy/src/AI/agent/[decisionManager.py](#)

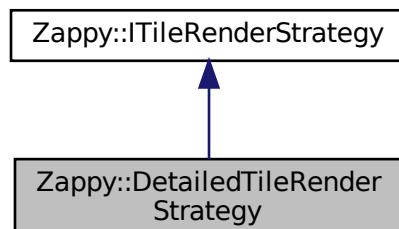
13.19 Zappy::DetailedTileRenderStrategy Class Reference

```
#include <DetailedTileRenderStrategy.hpp>
```

Inheritance diagram for Zappy::DetailedTileRenderStrategy:



Collaboration diagram for Zappy::DetailedTileRenderStrategy:



Public Member Functions

- `DetailedTileRenderStrategy (const std::shared_ptr< GraphicalContext > &ctx)`
- `void renderTile (const std::shared_ptr< IGraphicsLib > &graphicsLib, int x, int y, const ZappyTypes::Color &color, float tileSize, float spacing) override`

13.19.1 Detailed Description

Definition at line 16 of file DetailedTileRenderStrategy.hpp.

13.19.2 Constructor & Destructor Documentation

13.19.2.1 DetailedTileRenderStrategy()

```

Zappy::DetailedTileRenderStrategy::DetailedTileRenderStrategy (
    const std::shared_ptr< GraphicalContext > & ctx ) [explicit]
Definition at line 16 of file DetailedTileRenderStrategy.cpp.
17      : context(ctx) {}
  
```

13.19.3 Member Function Documentation

13.19.3.1 renderTile()

```
void Zappy::DetailedTileRenderStrategy::renderTile (
    const std::shared_ptr< IGraphicsLib > & graphicsLib,
    int x,
    int y,
    const ZappyTypes::Color & color,
    float tileSize,
    float spacing ) [override], [virtual]
```

Implements [Zappy::ITileRenderStrategy](#).

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

```
23
24     float mapOffset = context->getMapWidth() / 2.0f;
25     ZappyTypes::Vector3 position = {
26         (x - mapOffset + 0.5f) * (tileSize + spacing),
27         0.0f,
28         (y - mapOffset + 0.5f) * (tileSize + spacing)
29     };
30     graphicsLib->DrawCube(position, tileSize, 0.1f, tileSize, color);
31     ZappyTypes::Color borderColor = {100, 100, 100, 255};
32     float offset = tileSize/2;
33     graphicsLib->DrawLine3D({position.x - offset, position.y + 0.05f, position.z - offset},
34                             {position.x + offset, position.y + 0.05f, position.z - offset},
35                             borderColor);
36     graphicsLib->DrawLine3D({position.x + offset, position.y + 0.05f, position.z - offset},
37                             {position.x + offset, position.y + 0.05f, position.z + offset},
38                             borderColor);
39     graphicsLib->DrawLine3D({position.x + offset, position.y + 0.05f, position.z + offset},
40                             {position.x - offset, position.y + 0.05f, position.z + offset},
41                             borderColor);
42     graphicsLib->DrawLine3D({position.x - offset, position.y + 0.05f, position.z + offset},
43                             {position.x - offset, position.y + 0.05f, position.z - offset},
44                             borderColor);
45     const TileData& tileData = context->getTileData(x, y);
46     for (int i = 0; i < static_cast<int>(ResourceType::COUNT); ++i) {
47         int quantity = tileData.resources[i];
48         if (quantity > 0) {
49             renderResourceIndicator(graphicsLib, position, static_cast<ResourceType>(i), quantity,
50                                     tileSize);
51         }
52     }
53     for (int playerId : tileData.playerIds) {
54         renderPlayerIndicator(graphicsLib, position, playerId, tileSize);
55     }
56     for (int eggId : tileData.eggIds) {
57         renderEggIndicator(graphicsLib, position, eggId, tileSize);
58     }
```

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

- [/root/Desktop/Zappy/src/GUI/renderer/strategies/DetailedTileRenderStrategy.hpp](#)
- [/root/Desktop/Zappy/src/GUI/renderer/strategies/DetailedTileRenderStrategy.cpp](#)

13.20 DLLoader< T > Class Template Reference

```
#include <DLLoader.hpp>
```

Public Member Functions

- [DLLoader](#) (const std::string &libPath)

Constructeur chargeant une bibliothèque dynamique.
- [~DLLoader](#) ()

Destructeur déchargeant la bibliothèque.
- template<typename SymbolType >
 [SymbolType getSymbol](#) (const std::string &symbolName)

Récupère un symbole (fonction ou variable) depuis la bibliothèque.

- std::unique_ptr< T, std::function< void(T *)> > **getInstance** (const std::string &creatorName)
Crée une instance de l'interface en utilisant une fonction de création.

13.20.1 Detailed Description

```
template<typename T>
class DLLoader< T >
```

Definition at line 17 of file DLLoader.hpp.

13.20.2 Constructor & Destructor Documentation

13.20.2.1 DLLoader()

```
template<typename T >
DLLoader< T >::DLLoader (
    const std::string & libPath ) [inline], [explicit]
```

Constructeur chargeant une bibliothèque dynamique.

Parameters

<i>libPath</i>	Chemin vers la bibliothèque à charger
----------------	---------------------------------------

Definition at line 23 of file DLLoader.hpp.

```
23
24     _handle = dlopen(libPath.c_str(), RTLD_LAZY);
25     if (!_handle) {
26         throw std::runtime_error("Impossible de charger la bibliothèque: " + std::string(dlerror()));
27     }
28 }
```

13.20.2.2 ~DLLoader()

```
template<typename T >
DLLoader< T >::~DLLoader () [inline]
```

Destructeur déchargeant la bibliothèque.

Definition at line 33 of file DLLoader.hpp.

```
33
34     if (_handle) {
35         dlclose(_handle);
36     }
37 }
```

13.20.3 Member Function Documentation

13.20.3.1 getInstance()

```
template<typename T >
std::unique_ptr<T, std::function<void(T*)> > DLLoader< T >::getInstance (
    const std::string & creatorName ) [inline]
```

Crée une instance de l'interface en utilisant une fonction de création.

Parameters

<i>creatorName</i>	Nom de la fonction de création dans la bibliothèque
--------------------	---

Returns

Pointeur vers l'instance créée avec un deleteur personnalisé

Definition at line 58 of file DLLoader.hpp.

```

58
59     try {
60         using SharedCreatorFunc = std::shared_ptr<T> (*)();
61         SharedCreatorFunc sharedCreator = getSymbol<SharedCreatorFunc>(creatorName);
62         std::shared_ptr<T> sharedInstance = sharedCreator();
63         if (!sharedInstance)
64             throw std::runtime_error("La fonction de création a retourné un pointeur null");
65         return std::unique_ptr<T, std::function<void(T*)>>(
66             sharedInstance.get(),
67             [](T* /*ptr*/) { } // Utilisation de commentaire pour indiquer un paramètre
intentionnellement non utilisé
68         );
69     } catch (const std::exception& e) {
70         using RawCreatorFunc = T* (*)();
71         RawCreatorFunc rawCreator = getSymbol<RawCreatorFunc>(creatorName);
72         T* instance = rawCreator();
73         if (!instance)
74             throw std::runtime_error("La fonction de création a retourné un pointeur null");
75         return std::unique_ptr<T, std::function<void(T*)>>(instance, std::default_delete<T>());
76     }
77 }
```

13.20.3.2 getSymbol()

```

template<typename T >
template<typename SymbolType >
SymbolType DLLoader< T >::getSymbol (
    const std::string & symbolName ) [inline]
Récupère un symbole (fonction ou variable) depuis la bibliothèque.
```

Parameters

<i>symbolName</i>	Nom du symbole à charger
-------------------	--------------------------

Returns

Pointeur vers le symbole chargé

Definition at line 45 of file DLLoader.hpp.

```

45
46     void* symbol = dlsym(_handle, symbolName.c_str());
47     if (!symbol) {
48         throw std::runtime_error("Impossible de trouver le symbole '" + symbolName + "': " +
        std::string(dlerror()));
49     }
50     return reinterpret_cast<SymbolType>(symbol);
51 }
```

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

- /root/Desktop/Zappy/src/Shared/DLLoader.hpp

13.21 RayGUICPP::DropdownBox Class Reference

```
#include <RayGuiEncap.hpp>
```

Static Public Member Functions

- static int [Draw](#) (const Rectangle &bounds, const std::string &text, int active, bool editMode)

13.21.1 Detailed Description

Definition at line 86 of file RayGuiEncap.hpp.

13.21.2 Member Function Documentation

13.21.2.1 Draw()

```
static int RayGUICPP::DropdownBox::Draw (
    const Rectangle & bounds,
    const std::string & text,
    int active,
    bool editMode ) [inline], [static]
```

Definition at line 88 of file RayGuiEncap.hpp.

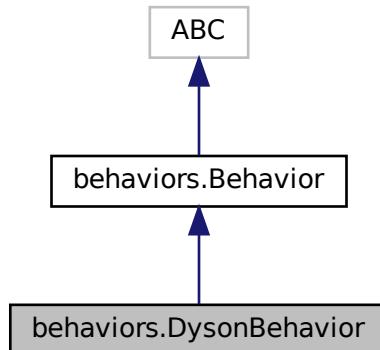
```
88
89     int value = active;
90     GuiDropdownBox(bounds, text.c_str(), &value, editMode);
91     return value;
92 }
```

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

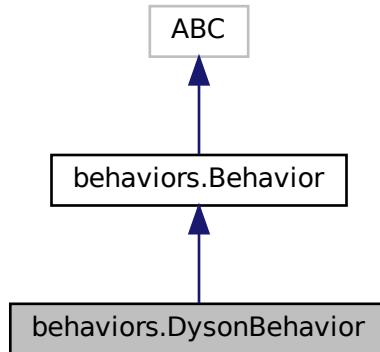
- /root/Desktop/Zappy/libs/RayGUICPP/include/RayGuiEncap.hpp

13.22 behaviors.DysonBehavior Class Reference

Inheritance diagram for behaviors.DysonBehavior:



Collaboration diagram for behaviors.DysonBehavior:



Public Member Functions

- def `execute` (self, surroundings=None, inventory=None)

Additional Inherited Members

13.22.1 Detailed Description

Definition at line 59 of file behaviors.py.

13.22.2 Member Function Documentation

13.22.2.1 execute()

```
def behaviors.DysonBehavior.execute (
    self,
    surroundings = None,
    inventory = None )
```

Reimplemented from [behaviors.Behavior](#).

Definition at line 60 of file behaviors.py.

```
60     def execute(self, surroundings=None, inventory=None):
61         for _ in range(self.agent.map_size_y):
62             actions.take_everything_here(self.agent)
63             self.agent.send_command("Forward")
64             self.agent.send_command("Right")
65             self.agent.send_command("Forward")
66             self.agent.send_command("Left")
67
68
```

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

- /root/Desktop/Zappy/src/AI/agent/[behaviors.py](#)

13.23 egg_s Struct Reference

```
#include <egg.h>
```

Collaboration diagram for egg_s:



Data Fields

- int `egg_id`
- char * `team_name`
- int `pos_x`
- int `pos_y`
- struct `egg_s` * `next`

13.23.1 Detailed Description

Definition at line 14 of file egg.h.

13.23.2 Field Documentation

13.23.2.1 `egg_id`

```
int egg_s::egg_id
```

Definition at line 15 of file egg.h.

13.23.2.2 `next`

```
struct egg_s* egg_s::next
```

Definition at line 19 of file egg.h.

13.23.2.3 `pos_x`

```
int egg_s::pos_x
```

Definition at line 17 of file egg.h.

13.23.2.4 `pos_y`

```
int egg_s::pos_y
```

Definition at line 18 of file egg.h.

13.23.2.5 `team_name`

```
char* egg_s::team_name
```

Definition at line 16 of file egg.h.

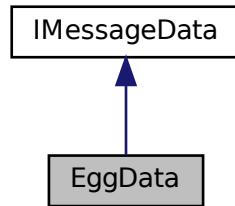
The documentation for this struct was generated from the following file:

- /root/Desktop/Zappy/src/Server/include/egg.h

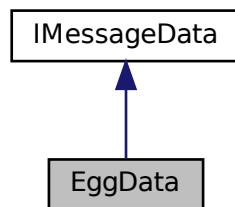
13.24 EggData Class Reference

```
#include <EggData.hpp>
```

Inheritance diagram for EggData:



Collaboration diagram for EggData:



Public Types

- enum class [EggAction](#) { Drop , Connection , Death , Laying }

Public Member Functions

- [EggData \(\)](#)
- [EggData \(int eggId, int playerId, int x, int y, EggAction action=EggAction::Drop\)](#)
- [MessageType getType \(\) const override](#)
- [EggAction getAction \(\) const](#)
- [int getEggId \(\) const](#)
- [int getPlayerId \(\) const](#)
- [int getX \(\) const](#)
- [int getY \(\) const](#)
- [void setEggId \(int value\)](#)
- [void setPlayerId \(int value\)](#)
- [void setX \(int value\)](#)
- [void setY \(int value\)](#)

13.24.1 Detailed Description

Definition at line 13 of file EggData.hpp.

13.24.2 Member Enumeration Documentation

13.24.2.1 EggAction

```
enum EggData::EggAction [strong]
```

Enumerator

Drop	
Connection	
Death	
Laying	

Definition at line 15 of file EggData.hpp.

```
15           {
16           Drop,      // enw - Egg creation
17           Connection, // ebo - Egg hatching/connection
18           Death,     // edi - Egg death
19           Laying     // pfk - Player starts laying an egg
20       };
```

13.24.3 Constructor & Destructor Documentation

13.24.3.1 EggData() [1/2]

```
EggData::EggData () [inline]
```

Definition at line 22 of file EggData.hpp.

```
22 : _eggId(0), _playerId(0), _x(0), _y(0), _action(EggAction::Drop) {}
```

13.24.3.2 EggData() [2/2]

```
EggData::EggData (
    int eggId,
    int playerId,
    int x,
    int y,
    EggAction action = EggAction::Drop ) [inline]
```

Definition at line 23 of file EggData.hpp.

```
24 : _eggId(eggId), _playerId(playerId), _x(x), _y(y), _action(action) {}
```

13.24.4 Member Function Documentation

13.24.4.1 getAction()

```
EggAction EggData::getAction () const [inline]
```

Definition at line 27 of file EggData.hpp.

```
27 { return _action; }
```

13.24.4.2 getEggId()

```
int EggData::getEggId ( ) const [inline]  
Definition at line 29 of file EggData.hpp.  
29 { return _eggId; }
```

13.24.4.3 getPlayerId()

```
int EggData::getPlayerId ( ) const [inline]  
Definition at line 30 of file EggData.hpp.  
30 { return _playerId; }
```

13.24.4.4 getType()

```
MessageType EggData::getType ( ) const [inline], [override], [virtual]  
Implements IMessageData.  
Definition at line 26 of file EggData.hpp.  
26 { return MessageType::Egg; }
```

13.24.4.5 getX()

```
int EggData::getX ( ) const [inline]  
Definition at line 31 of file EggData.hpp.  
31 { return _x; }
```

13.24.4.6 getY()

```
int EggData::getY ( ) const [inline]  
Definition at line 32 of file EggData.hpp.  
32 { return _y; }
```

13.24.4.7 setEggId()

```
void EggData::setEggId (  
    int value ) [inline]  
Definition at line 34 of file EggData.hpp.  
34 { _eggId = value; }
```

13.24.4.8 setPlayerId()

```
void EggData::setPlayerId (  
    int value ) [inline]  
Definition at line 35 of file EggData.hpp.  
35 { _playerId = value; }
```

13.24.4.9 setX()

```
void EggData::setX (  
    int value ) [inline]  
Definition at line 36 of file EggData.hpp.  
36 { _x = value; }
```

13.24.4.10 setY()

```
void EggData::setY (
    int value ) [inline]
Definition at line 37 of file EggData.hpp.
37 { _y = value; }
```

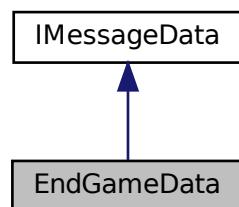
The documentation for this class was generated from the following file:

- /root/Desktop/Zappy/src/GUI/network/protocol/messageData/[EggData.hpp](#)

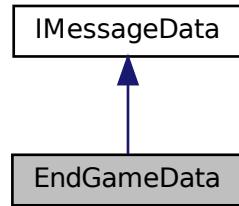
13.25 EndGameData Class Reference

```
#include <EndGameData.hpp>
```

Inheritance diagram for EndGameData:



Collaboration diagram for EndGameData:



Public Member Functions

- [EndGameData](#) (const std::string &teamName)
- [MessageType getType \(\)](#) const override
- const std::string & [getTeamName \(\)](#) const
- void [setTeamName \(const std::string &value\)](#)

13.25.1 Detailed Description

Definition at line 14 of file EndGameData.hpp.

13.25.2 Constructor & Destructor Documentation

13.25.2.1 EndGameData()

```
EndGameData::EndGameData (
    const std::string & teamName ) [inline], [explicit]
Definition at line 16 of file EndGameData.hpp.
17     : _teamName(teamName) {}
```

13.25.3 Member Function Documentation

13.25.3.1 getTeamName()

```
const std::string& EndGameData::getTeamName () const [inline]
Definition at line 21 of file EndGameData.hpp.
21 { return _teamName; }
```

13.25.3.2 getType()

```
MessageType EndGameData::getType () const [inline], [override], [virtual]
Implements IMessageData.
Definition at line 19 of file EndGameData.hpp.
19 { return MessageType::EndGame; }
```

13.25.3.3 setTeamName()

```
void EndGameData::setTeamName (
    const std::string & value ) [inline]
Definition at line 23 of file EndGameData.hpp.
23 { _teamName = value; }
```

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

- /root/Desktop/Zappy/src/GUI/network/protocol/messageData/[EndGameData.hpp](#)

13.26 raylibcpp::Font Class Reference

```
#include <Font.hpp>
```

Public Member Functions

- [Font](#) (const std::string &path)
- [~Font](#) ()
- void [drawText](#) (const std::string &text, int x, int y, int size, Color color) const
- [::Font get](#) () const

13.26.1 Detailed Description

Definition at line 14 of file Font.hpp.

13.26.2 Constructor & Destructor Documentation

13.26.2.1 Font()

```
raylibcpp::Font::Font (
    const std::string & path ) [explicit]
Definition at line 13 of file Font.cpp.
13
14     font = LoadFont(path.c_str());
15 }
```

13.26.2.2 ~Font()

```
raylibcpp::Font::~Font ( )
Definition at line 17 of file Font.cpp.
17
18     UnloadFont(font);
19 }
```

13.26.3 Member Function Documentation

13.26.3.1 drawText()

```
void raylibcpp::Font::drawText (
    const std::string & text,
    int x,
    int y,
    int size,
    Color color ) const
Definition at line 21 of file Font.cpp.
22
23     DrawTextEx(font, text.c_str(), {static_cast<float>(x), static_cast<float>(y)},
24                 static_cast<float>(size), 1, color);
25 }
```

13.26.3.2 get()

```
Font raylibcpp::Font::get ( ) const
Definition at line 27 of file Font.cpp.
27
28     return font;
29 }
```

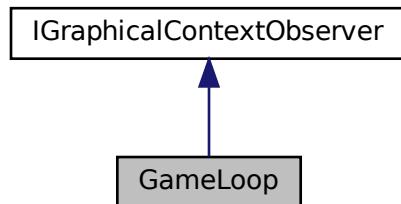
The documentation for this class was generated from the following files:

- /root/Desktop/Zappy/libs/RaylibCPP/font/Font.hpp
- /root/Desktop/Zappy/libs/RaylibCPP/font/Font.cpp

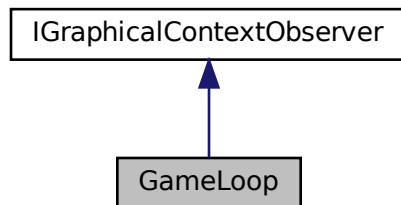
13.27 GameLoop Class Reference

```
#include <GameLoop.hpp>
```

Inheritance diagram for GameLoop:



Collaboration diagram for GameLoop:



Public Member Functions

- `GameLoop ()`
- `~GameLoop ()=default`
- `bool init ()`
- `int run ()`
- `void setServerInfo (const std::string &host, int port)`
- `void renderCube ()`
- `void onMapSizeChanged (int width, int height) override`
- `void onTileChanged (int x, int y, const TileData &tileData) override`

13.27.1 Detailed Description

Definition at line 21 of file GameLoop.hpp.

13.27.2 Constructor & Destructor Documentation

13.27.2.1 GameLoop()

```
GameLoop::GameLoop ( )  
Definition at line 17 of file GameLoop.cpp.  
18     : m_host("localhost"), m_port(4242) {  
19 }
```

13.27.2.2 ~GameLoop()

```
GameLoop::~GameLoop ( ) [default]
```

13.27.3 Member Function Documentation

13.27.3.1 init()

```
bool GameLoop::init ( )
Definition at line 21 of file GameLoop.cpp.
21
22     if (!loadLibraries())
23         return false;
24     initializeManagers();
25     if (!loadModels())
26         return false;
27     setupComponents();
28     return true;
29 }
```

13.27.3.2 onMapSizeChanged()

```
void GameLoop::onMapSizeChanged (
    int width,
    int height ) [override], [virtual]
Implements IGraphicalContextObserver.
```

Definition at line 108 of file GameLoop.cpp.

```
108
109     m_mapWidth = width;
110     m_mapHeight = height;
111     if (m_camera) {
112         m_camera->setMapDimensions(width, height);
113     }
114 }
```

13.27.3.3 onTileChanged()

```
void GameLoop::onTileChanged (
    int x,
    int y,
    const TileData & tileData ) [override], [virtual]
Implements IGraphicalContextObserver.
```

Definition at line 116 of file GameLoop.cpp.

```
116
117     //TODO(Sam): Implement tile change handling
118 }
```

13.27.3.4 renderCube()

```
void GameLoop::renderCube ( )
```

Definition at line 97 of file GameLoop.cpp.

```
97
98     auto& modelManager = ModelManager::getInstance();
99     if (m_cubeModelId != -1) {
100         ZappyTypes::Vector3 cubePosition = {2.0f, 0.0f, 2.0f};
101         ZappyTypes::Color cubeColor = {255, 255, 255, 255};
102         modelManager.drawModel(m_cubeModelId, cubePosition, cubeColor);
103     } else {
104         std::cerr << "Modèle cube.obj non trouvé." << std::endl;
105     }
106 }
```

13.27.3.5 run()

```
int GameLoop::run ( )
Definition at line 70 of file GameLoop.cpp.
70      {
71      if (!m_graphics || !m_gui || !m_renderer || !m_camera || !m_uiRenderer) {
72          std::cerr << "Game components not initialized properly." << std::endl;
73          return 84;
74      }
75
76      while (!m_graphics->WindowShouldClose ()) {
77          m_camera->update (m_graphics);
78          m_graphics->BeginDrawing ();
79          m_graphics->ClearBackground ({32, 32, 64, 255});
80          m_graphics->BeginCamera3D ();
81          m_graphics->DrawPlane ({0.0f, 0.0f, 0.0f}, {20.0f, 20.0f}, {200, 200, 200, 255});
82          renderCube ();
83          m_graphics->EndCamera3D ();
84          m_uiRenderer->renderUI (m_graphics, m_gui, m_camera);
85          m_graphics->EndDrawing ();
86          std::this_thread::sleep_for (std::chrono::milliseconds (16));
87      }
88      m_graphics->CloseWindow ();
89      return 0;
90 }
```

13.27.3.6 setServerInfo()

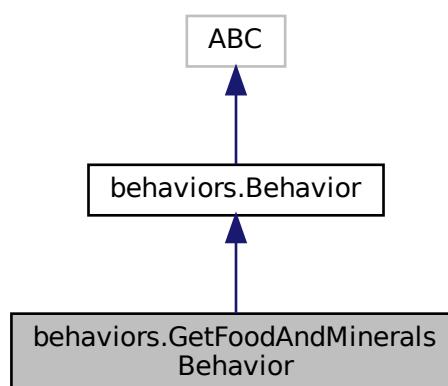
```
void GameLoop::setServerInfo (
    const std::string & host,
    int port )
Definition at line 92 of file GameLoop.cpp.
92
93     m_host = host;
94     m_port = port;
95 }
```

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

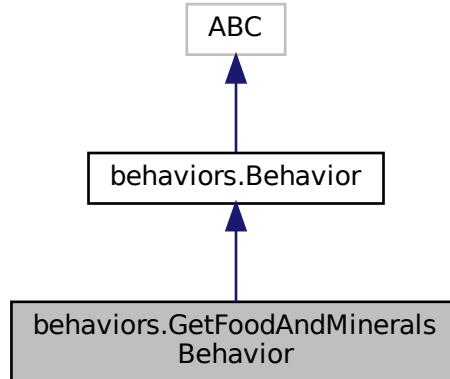
- /root/Desktop/Zappy/src/GUI/GameLoop.hpp
- /root/Desktop/Zappy/src/GUI/GameLoop.cpp

13.28 behaviors.GetFoodAndMineralsBehavior Class Reference

Inheritance diagram for behaviors.GetFoodAndMineralsBehavior:



Collaboration diagram for behaviors.GetFoodAndMineralsBehavior:



Public Member Functions

- def `execute` (self, surroundings=None, inventory=None)

Additional Inherited Members

13.28.1 Detailed Description

Definition at line 69 of file behaviors.py.

13.28.2 Member Function Documentation

13.28.2.1 execute()

```
def behaviors.GetFoodAndMineralsBehavior.execute (
    self,
    surroundings = None,
    inventory = None )
```

Reimplemented from `behaviors.Behavior`.

Definition at line 70 of file behaviors.py.

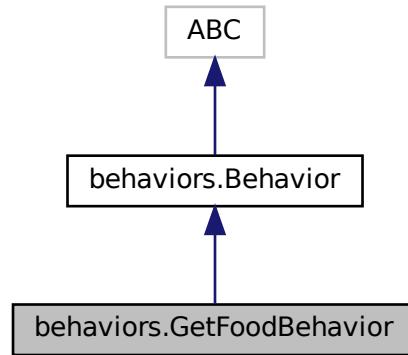
```
70     def execute(self, surroundings=None, inventory=None):
71         if not surroundings or not inventory:
72             print("GetFoodAndMineralsBehaviour: Surroundings or inventory is None.")
73             return
74
75         if zappy.parse_inventory(inventory).get("food", 0) < 10:
76             GetFoodBehavior(self.agent).execute(surroundings, inventory)
77         else:
78             GetMineralsBehavior(self.agent).execute(surroundings, inventory)
```

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

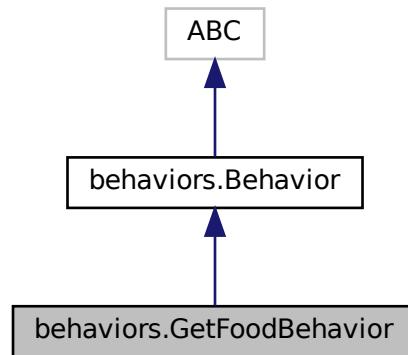
- /root/Desktop/Zappy/src/AI/agent/[behaviors.py](#)

13.29 behaviors.GetFoodBehavior Class Reference

Inheritance diagram for behaviors.GetFoodBehavior:



Collaboration diagram for behaviors.GetFoodBehavior:



Public Member Functions

- def `execute` (self, surroundings=None, `inventory`=None)

Additional Inherited Members

13.29.1 Detailed Description

Definition at line 16 of file behaviors.py.

13.29.2 Member Function Documentation

13.29.2.1 execute()

```
def behaviors.GetFoodBehavior.execute (
    self,
    surroundings = None,
    inventory = None )
```

Reimplemented from [behaviors.Behavior](#).

Definition at line 17 of file [behaviors.py](#).

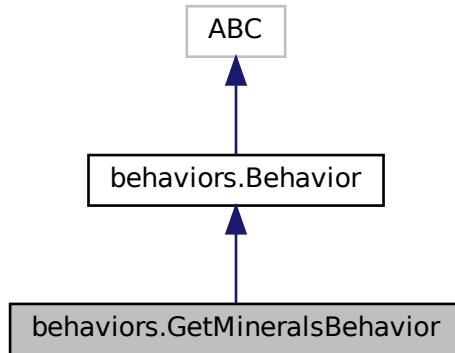
```
17     def execute(self, surroundings=None, inventory=None):
18         actions.go_take_item(self.agent, "food")
19
20
```

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

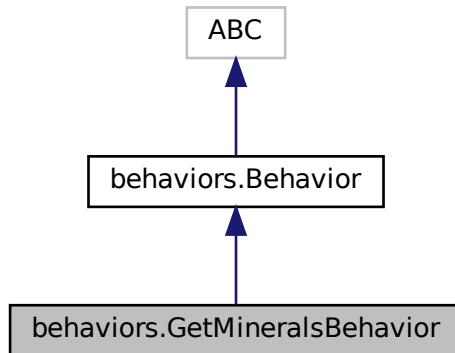
- /root/Desktop/Zappy/src/AI/agent/[behaviors.py](#)

13.30 behaviors.GetMineralsBehavior Class Reference

Inheritance diagram for behaviors.GetMineralsBehavior:



Collaboration diagram for behaviors.GetMineralsBehavior:



Public Member Functions

- def `execute` (self, surroundings=None, inventory=None)

Additional Inherited Members

13.30.1 Detailed Description

Definition at line 51 of file behaviors.py.

13.30.2 Member Function Documentation

13.30.2.1 execute()

```
def behaviors.GetMineralsBehavior.execute (
    self,
    surroundings = None,
    inventory = None )
```

Reimplemented from `behaviors.Behavior`.

Definition at line 52 of file behaviors.py.

```
52     def execute(self, surroundings=None, inventory=None):
53         if not surroundings:
54             print("GetMineralsBehavior: Surroundings is None.")
55             return
56         actions.go_take_item(self.agent, zappy.get_best_available_resource(surroundings))
57
58
```

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

- /root/Desktop/Zappy/src/AI/agent/`behaviors.py`

13.31 raylibcpp::GlyphDimensions Struct Reference

```
#include <Text3DHelper.hpp>
```

Data Fields

- float `width`
- float `height`

13.31.1 Detailed Description

Definition at line 15 of file Text3DHelper.hpp.

13.31.2 Field Documentation

13.31.2.1 height

```
float raylibcpp::GlyphDimensions::height
```

Definition at line 17 of file Text3DHelper.hpp.

13.31.2.2 width

```
float raylibcpp::GlyphDimensions::width
```

Definition at line 16 of file Text3DHelper.hpp.

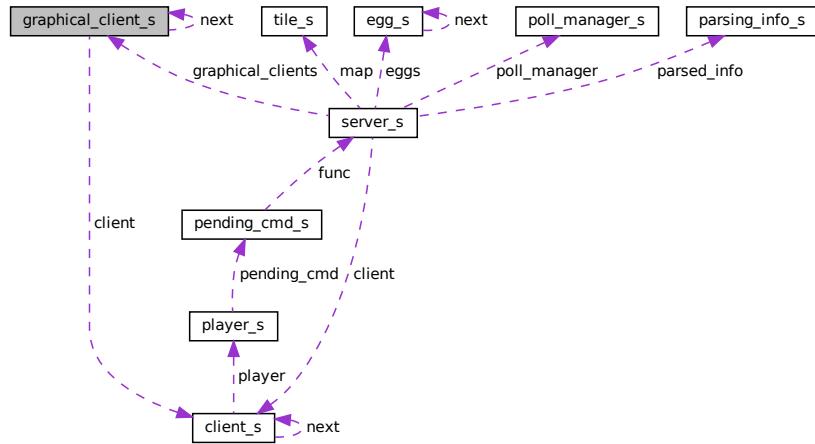
The documentation for this struct was generated from the following file:

- /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/`Text3DHelper.hpp`

13.32 graphical_client_s Struct Reference

```
#include <client.h>
```

Collaboration diagram for graphical_client_s:



Data Fields

- `client_t * client`
- `struct graphical_client_s * next`

13.32.1 Detailed Description

Definition at line 19 of file client.h.

13.32.2 Field Documentation

13.32.2.1 client

```
client_t* graphical_client_s::client
```

Definition at line 20 of file client.h.

13.32.2.2 next

```
struct graphical_client_s* graphical_client_s::next
```

Definition at line 21 of file client.h.

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

- /root/Desktop/Zappy/src/Server/include/client.h

13.33 GraphicalContext Class Reference

```
#include <GraphicalContext.hpp>
```

Public Types

- `typedef void(GraphicalContext::* UpdateFunction) (std::shared_ptr< IMessageData >)`

Public Member Functions

- `GraphicalContext ()`
- `~GraphicalContext ()=default`
- `void updateContext (const Message &message)`
Ajoute un observateur au contexte graphique.
- `void addObserver (std::shared_ptr< IGraphicalContextObserver > observer)`
Supprime un observateur du contexte graphique.
- `int getMapWidth () const`
Récupère la largeur de la carte.
- `int getMapHeight () const`
Récupère la hauteur de la carte.
- `const TileData & getTileData (int x, int y) const`
Récupère les données d'une tuile.
- `bool isPlayerOnTile (int x, int y, int playerId) const`
Vérifie si un joueur est présent sur une tuile.
- `int getResourceQuantity (int x, int y, ResourceType resourceType) const`
Récupère la quantité d'une ressource sur une tuile.
- `ResourceType getDominantResourceType (int x, int y) const`
Récupère le type de ressource dominant sur une tuile.
- `const std::shared_ptr< PlayerInfoData > getPlayerInfo (int playerId) const`
Récupère les informations d'un joueur.

13.33.1 Detailed Description

Definition at line 58 of file GraphicalContext.hpp.

13.33.2 Member Typedef Documentation

13.33.2.1 UpdateFunction

```
typedef void(GraphicalContext::* GraphicalContext::UpdateFunction) (std::shared_ptr< IMESSAGEData >)
```

Definition at line 60 of file GraphicalContext.hpp.

13.33.3 Constructor & Destructor Documentation

13.33.3.1 GraphicalContext()

```
GraphicalContext::GraphicalContext ( )
```

Definition at line 14 of file GraphicalContext.cpp.

```
14
15     _updateFunctions = {
16         {MessageType::MapSize, &GraphicalContext::updateMapSize},
17         {MessageType::TileContent, &GraphicalContext::updateTileContent},
18         {MessageType::TeamName, &GraphicalContext::updateTeamName},
19         {MessageType::PlayerInfo, &GraphicalContext::updatePlayerInfo},
20         {MessageType::PlayerInventory, &GraphicalContext::updatePlayerInventory},
21         {MessageType::Broadcast, &GraphicalContext::updatePlayerBroadcast},
22         {MessageType::Incantation, &GraphicalContext::updateIncantationStart},
23         {MessageType::IncantationEnd, &GraphicalContext::updateIncantationEnd},
24         {MessageType::TimeUnit, &GraphicalContext::updateTimeUnit},
25         {MessageType::EndGame, &GraphicalContext::updateEndGame},
26         {MessageType::ServerMessage, &GraphicalContext::updateServerMessage},
27         {MessageType::Egg, &GraphicalContext::updateEggDrop}
28     };
29 }
```

13.33.3.2 ~GraphicalContext()

```
GraphicalContext::~GraphicalContext () [default]
```

13.33.4 Member Function Documentation

13.33.4.1 addObserver()

```
void GraphicalContext::addObserver (
    std::shared_ptr< IGraphicalContextObserver > observer )
```

Ajoute un observateur au contexte graphique.

Parameters

<i>observer</i>	L'observateur à ajouter
-----------------	-------------------------

Definition at line 263 of file GraphicalContext.cpp.

```
263
264     auto it = std::find_if(_observers.begin(), _observers.end(),
265     [&](const std::shared_ptr<IGraphicalContextObserver>& obs) {
266         return obs == observer;
267     });
268     if (it == _observers.end()) {
269         _observers.push_back(observer);
270         if (_mapWidth > 0 && _mapHeight > 0) {
271             observer->onMapSizeChanged(_mapWidth, _mapHeight);
272             for (int y = 0; y < _mapHeight; ++y) {
273                 for (int x = 0; x < _mapWidth; ++x) {
274                     observer->onTileChanged(x, y, _mapTiles[y][x]);
275                 }
276             }
277         }
278     }
279 }
```

13.33.4.2 getDominantResourceType()

```
ResourceType GraphicalContext::getDominantResourceType (
    int x,
    int y ) const
```

Récupère le type de ressource dominant sur une tuile.

Parameters

<i>x</i>	Coordonnée X de la tuile
<i>y</i>	Coordonnée Y de la tuile

Returns

Le type de ressource dominant

Definition at line 310 of file GraphicalContext.cpp.

```
310
311     if (!isValidCoordinates(x, y))
312         return ResourceType::COUNT;
313     const TileData& tile = _mapTiles[y][x];
314     ResourceType dominantType = ResourceType::COUNT;
315     int maxQuantity = 0;
316     for (int i = 0; i < static_cast<int>(ResourceType::COUNT); ++i) {
317         if (tile.resources[i] > maxQuantity) {
318             maxQuantity = tile.resources[i];
319             dominantType = static_cast<ResourceType>(i);
320         }
321     }
322     if (maxQuantity == 0)
323         return ResourceType::COUNT;
```

```
324     return dominantType;
325 }
```

13.33.4.3 getMapHeight()

int GraphicalContext::getMapHeight () const [inline]
Récupère la hauteur de la carte.

Returns

La hauteur de la carte

Definition at line 88 of file GraphicalContext.hpp.
88 { return _mapHeight; }

13.33.4.4 getMapWidth()

int GraphicalContext::getMapWidth () const [inline]
Récupère la largeur de la carte.

Returns

La largeur de la carte

Definition at line 82 of file GraphicalContext.hpp.
82 { return _mapWidth; }

13.33.4.5 getPlayerInfo()

const std::shared_ptr< PlayerInfoData > GraphicalContext::getPlayerInfo (
 int playerId) const
Récupère les informations d'un joueur.

Parameters

<i>playerId</i>	ID du joueur
-----------------	--------------

Returns

Les informations du joueur ou nullptr si le joueur n'existe pas

Definition at line 347 of file GraphicalContext.cpp.

```
347
348     return (_players.find(playerId) != _players.end()) ?
349             std::make_shared<PlayerInfoData>(_players.at(playerId)) : nullptr;
350 }
```

13.33.4.6 getResourceQuantity()

```
int GraphicalContext::getResourceQuantity (
    int x,
    int y,
    ResourceType resourceType ) const
```

Récupère la quantité d'une ressource sur une tuile.

Parameters

<i>x</i>	Coordonnée X de la tuile
<i>y</i>	Coordonnée Y de la tuile
<i>resourceType</i>	Type de ressource

Returns

La quantité de ressource

Definition at line 334 of file GraphicalContext.cpp.

```
334
335     if (!isValidCoordinates(x, y) || resourceType == ResourceType::COUNT)
336         return 0;
337     return _mapTiles[y][x].resources[static_cast<int>(resourceType)];
338 }
```

{

13.33.4.7 getTileData()

```
const TileData & GraphicalContext::getTileData (
    int x,
    int y ) const
```

Récupère les données d'une tuile.

Parameters

<i>x</i>	Coordonnée X de la tuile
<i>y</i>	Coordonnée Y de la tuile

Returns

Les données de la tuile

Definition at line 327 of file GraphicalContext.cpp.

```
327
328     static TileData emptyTile;
329     if (!isValidCoordinates(x, y))
330         return emptyTile;
331     return _mapTiles[y][x];
332 }
```

{

13.33.4.8 isPlayerOnTile()

```
bool GraphicalContext::isPlayerOnTile (
    int x,
    int y,
    int playerId ) const
```

Vérifie si un joueur est présent sur une tuile.

Parameters

<i>x</i>	Coordonnée X de la tuile
<i>y</i>	Coordonnée Y de la tuile
<i>playerId</i>	ID du joueur

Returns

true si le joueur est présent, false sinon

Definition at line 340 of file GraphicalContext.cpp.

```
340
341     if (!isValidCoordinates(x, y))
342         return false;
343     const auto& playerIds = _mapTiles[y][x].playerIds;
344     return std::find(playerIds.begin(), playerIds.end(), playerId) != playerIds.end();
345 }
```

{

13.33.4.9 removeObserver()

```
void GraphicalContext::removeObserver (
    std::shared_ptr< IGraphicalContextObserver > observer )
```

Supprime un observateur du contexte graphique.

Parameters

<i>observer</i>	L'observateur à supprimer
-----------------	---------------------------

Definition at line 281 of file GraphicalContext.cpp.

```
281
282     _observers.erase(
283         std::remove_if(_observers.begin(), _observers.end(),
284             [&] (const std::shared_ptr<IGraphicalContextObserver>& obs) {
285                 return obs == observer;
286             }
287         ),
288         _observers.end()
289     );
290 }
```

13.33.4.10 updateContext()

```
void GraphicalContext::updateContext (
    const Message & message )
```

Definition at line 32 of file GraphicalContext.cpp.

```
32
33     MessageType data;
34
35     if (!message.getStructuredData()) {
36         std::cerr << "Received message with no structured data" << std::endl;
37         return;
38     }
39     data = message.getStructuredData() ->getType();
40     auto it = _updateFunctions.find(data);
41     if (it != _updateFunctions.end()) {
42         UpdateFunction updateFunc = it->second;
43         (this->*updateFunc)(message.getStructuredData());
44     } else {
45         std::cerr << "No update function for message type: " << static_cast<int>(data) << std::endl;
46     }
47 }
```

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

- /root/Desktop/Zappy/src/GUI/graphicalContext/GraphicalContext.hpp
- /root/Desktop/Zappy/src/GUI/graphicalContext/GraphicalContext.cpp

13.34 RayGUICPP::GroupBox Class Reference

```
#include <RayGuiEncap.hpp>
```

Static Public Member Functions

- static void **Draw** (const Rectangle &bounds, const std::string &text)

13.34.1 Detailed Description

Definition at line 166 of file RayGuiEncap.hpp.

13.34.2 Member Function Documentation

13.34.2.1 Draw()

```
static void RayGUICPP::GroupBox::Draw (
    const Rectangle & bounds,
    const std::string & text ) [inline], [static]
Definition at line 168 of file RayGuiEncap.hpp.
168
169     GuiGroupBox(bounds, text.c_str());
170 }
```

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

- /root/Desktop/Zappy/libs/RayGUICPP/include/RayGuiEncap.hpp

13.35 RayGUICPP::Icon Class Reference

```
#include <RayGuiEncap.hpp>
```

Static Public Member Functions

- static std::string [IconText](#) (int iconId, const std::string &text)

13.35.1 Detailed Description

Definition at line 246 of file RayGuiEncap.hpp.

13.35.2 Member Function Documentation

13.35.2.1 [IconText\(\)](#)

```
static std::string RayGUICPP::Icon::IconText (
    int iconId,
    const std::string & text ) [inline], [static]
Definition at line 248 of file RayGuiEncap.hpp.
248
249     return GuiIconText(iconId, text.c_str());
250 }
```

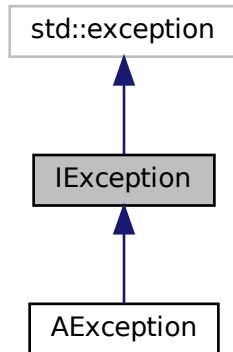
The documentation for this class was generated from the following file:

- /root/Desktop/Zappy/libs/RayGUICPP/include/RayGuiEncap.hpp

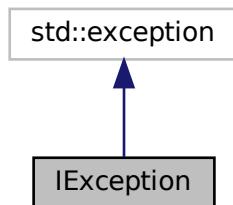
13.36 IException Class Reference

```
#include <IException.hpp>
```

Inheritance diagram for IException:



Collaboration diagram for IException:



Public Member Functions

- virtual `~IException () noexcept=default`
- const char * `what () const noexcept override=0`
- virtual std::string `getMessage () const noexcept=0`
- virtual std::string `getType () const noexcept=0`
- virtual std::string `getFormattedMessage () const noexcept=0`

13.36.1 Detailed Description

Definition at line 15 of file IException.hpp.

13.36.2 Constructor & Destructor Documentation

13.36.2.1 ~IException()

```
virtual IException::~IException ( ) [virtual], [default], [noexcept]
```

13.36.3 Member Function Documentation

13.36.3.1 `getFormattedMessage()`

```
virtual std::string IException::getFormattedMessage ( ) const [pure virtual], [noexcept]
Implemented in AException.
```

13.36.3.2 `getMessage()`

```
virtual std::string IException::getMessage ( ) const [pure virtual], [noexcept]
Implemented in AException.
```

13.36.3.3 `getType()`

```
virtual std::string IException::getType ( ) const [pure virtual], [noexcept]
Implemented in AException.
```

13.36.3.4 `what()`

```
const char* IException::what ( ) const [override], [pure virtual], [noexcept]
Implemented in AException.
```

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

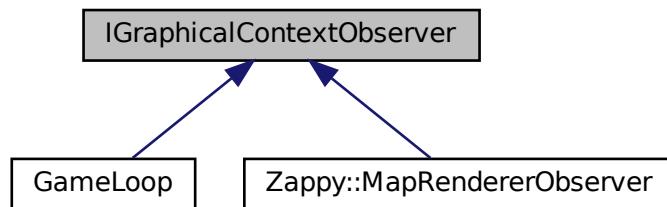
- /root/Desktop/Zappy/src/GUI/shared/exception/[IException.hpp](#)

13.37 IGraphicalContextObserver Class Reference

Observateur de changements de contexte graphique.

```
#include <GraphicalContext.hpp>
```

Inheritance diagram for IGraphicalContextObserver:



Public Member Functions

- virtual [~IGraphicalContextObserver](#) ()=default
- virtual void [onMapSizeChanged](#) (int width, int height)=0
- virtual void [onTileChanged](#) (int x, int y, const [TileData](#) &tileData)=0

13.37.1 Detailed Description

Observateur de changements de contexte graphique.
Definition at line 36 of file GraphicalContext.hpp.

13.37.2 Constructor & Destructor Documentation

13.37.2.1 ~IGraphicalContextObserver()

```
virtual IGraphicalContextObserver::~IGraphicalContextObserver () [virtual], [default]
```

13.37.3 Member Function Documentation

13.37.3.1 onMapSizeChanged()

```
virtual void IGraphicalContextObserver::onMapSizeChanged (
    int width,
    int height ) [pure virtual]
```

Implemented in [Zappy::MapRendererObserver](#), and [GameLoop](#).

13.37.3.2 onTileChanged()

```
virtual void IGraphicalContextObserver::onTileChanged (
    int x,
    int y,
    const TileData & tileData ) [pure virtual]
```

Implemented in [Zappy::MapRendererObserver](#), and [GameLoop](#).

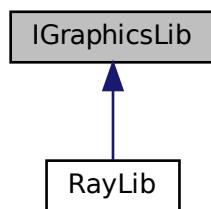
The documentation for this class was generated from the following file:

- /root/Desktop/Zappy/src/GUI/graphicalContext/[GraphicalContext.hpp](#)

13.38 IGraphicsLib Class Reference

```
#include <IGraphicsLib.hpp>
```

Inheritance diagram for IGraphicsLib:



Public Member Functions

- virtual `~IGraphicsLib ()=default`
- virtual void `InitWindow (int width, int height, const std::string &title)=0`
- virtual void `CloseWindow ()=0`
- virtual bool `WindowShouldClose ()=0`
- virtual void `setFps (int fps)=0`
- virtual void `BeginDrawing ()=0`
- virtual void `EndDrawing ()=0`
- virtual void `ClearBackground (ZappyTypes::Color color)=0`
- virtual void `DrawRectangle (int x, int y, int width, int height, ZappyTypes::Color color)=0`
- virtual void `DrawCircle (int centerX, int centerY, float radius, ZappyTypes::Color color)=0`
- virtual void `DrawCube (ZappyTypes::Vector3 position, float width, float height, float length, ZappyTypes::Color color)=0`
- virtual void `DrawSphere (ZappyTypes::Vector3 centerPos, float radius, ZappyTypes::Color color)=0`
- virtual void `DrawPlane (ZappyTypes::Vector3 centerPos, ZappyTypes::Vector2 size, ZappyTypes::Color color)=0`
- virtual void `DrawGrid (int slices, float spacing)=0`
- virtual void `DrawCylinder (ZappyTypes::Vector3 position, float radiusTop, float radiusBottom, float height, int slices, ZappyTypes::Color color)=0`
- virtual void `DrawLine3D (ZappyTypes::Vector3 startPos, ZappyTypes::Vector3 endPos, ZappyTypes::Color color)=0`
- virtual int `LoadTexture2D (const std::string &path)=0`
- virtual void `DrawTexture2D (int textureId, int x, int y)=0`
- virtual void `UnloadTexture2D (int textureId)=0`
- virtual bool `IsTextureReady (int textureId) const =0`
- virtual void `LoadFont (const std::string &path)=0`
- virtual void `DrawText (const std::string &text, int x, int y, int size, ZappyTypes::Color color)=0`
- virtual void `UnloadFont ()=0`
- virtual void `DrawText3D (const std::string &text, ZappyTypes::Vector3 position, float fontSize, float fontSpacing, float lineSpacing, bool backface, ZappyTypes::Color tint)=0`
- virtual void `DrawTextWave3D (const std::string &text, ZappyTypes::Vector3 position, float fontSize, float fontSpacing, float lineSpacing, bool backface, float time, ZappyTypes::Color tint)=0`
- virtual `ZappyTypes::Vector3 MeasureText3D (const std::string &text, float fontSize, float fontSpacing, float lineSpacing)=0`
- virtual bool `IsKeyPressed (int key)=0`
- virtual bool `IsKeyDown (int key)=0`
- virtual bool `IsMouseButtonPressed (int button)=0`
- virtual bool `IsMouseButtonDown (int button)=0`
- virtual bool `IsMouseButtonReleased (int button)=0`
- virtual int `GetMouseX ()=0`
- virtual int `GetMouseY ()=0`
- virtual `ZappyTypes::Vector2 GetMousePosition ()=0`
- virtual float `GetMouseWheelMove ()=0`
- virtual void `PlaySound (const std::string &path)=0`
- virtual void `StopSound ()=0`
- virtual void `SetSoundVolume (float volume)=0`
- virtual void `PlayMusic (const std::string &path)=0`
- virtual void `StopMusic ()=0`
- virtual void `SetMusicVolume (float volume)=0`
- virtual void `UpdateMusic ()=0`
- virtual void `CreateCamera2D ()=0`
- virtual void `CreateCamera3D (ZappyTypes::Vector3 position, ZappyTypes::Vector3 target, ZappyTypes::Vector3 up, float fovy, int projection)=0`
- virtual void `BeginCamera3D ()=0`
- virtual void `EndCamera3D ()=0`

- virtual void [LoadTexture3D](#) (const std::string &path)=0
- virtual void [BindTexture3D](#) (int unit=0)=0
- virtual void [UnloadTexture3D](#) ()=0
- virtual int [LoadModel3D](#) (const std::string &path)=0
- virtual void [DrawModel3D](#) (int modelId, ZappyTypes::Vector3 position, float scale, ZappyTypes::Color color)=0
- virtual void [DrawModelEx](#) (int modelId, ZappyTypes::Vector3 position, ZappyTypes::Vector3 rotationAxis, float rotationAngle, float scale)=0
- virtual void [UnloadModel3D](#) (int modelId)=0
- virtual int [LoadModelWithTexture](#) (const std::string &modelPath, const std::string &texturePath)=0

13.38.1 Detailed Description

Definition at line 13 of file IGraphicsLib.hpp.

13.38.2 Constructor & Destructor Documentation

13.38.2.1 ~IGraphicsLib()

```
virtual IGraphicsLib::~IGraphicsLib ( ) [virtual], [default]
```

13.38.3 Member Function Documentation

13.38.3.1 BeginCamera3D()

```
virtual void IGraphicsLib::BeginCamera3D ( ) [pure virtual]  
Implemented in RayLib.
```

13.38.3.2 BeginDrawing()

```
virtual void IGraphicsLib::BeginDrawing ( ) [pure virtual]  
Implemented in RayLib.
```

13.38.3.3 BindTexture3D()

```
virtual void IGraphicsLib::BindTexture3D (   
    int unit = 0 ) [pure virtual]  
Implemented in RayLib.
```

13.38.3.4 ClearBackground()

```
virtual void IGraphicsLib::ClearBackground (   
    ZappyTypes::Color color ) [pure virtual]  
Implemented in RayLib.
```

13.38.3.5 CloseWindow()

```
virtual void IGraphicsLib::CloseWindow ( ) [pure virtual]  
Implemented in RayLib.
```

13.38.3.6 CreateCamera2D()

```
virtual void IGraphicsLib::CreateCamera2D ( ) [pure virtual]  
Implemented in RayLib.
```

13.38.3.7 CreateCamera3D()

```
virtual void IGraphicsLib::CreateCamera3D (   
    ZappyTypes::Vector3 position,  
    ZappyTypes::Vector3 target,  
    ZappyTypes::Vector3 up,  
    float fovy,  
    int projection ) [pure virtual]
```

Implemented in RayLib.

13.38.3.8 DrawCircle()

```
virtual void IGraphicsLib::DrawCircle (   
    int centerX,  
    int centerY,  
    float radius,  
    ZappyTypes::Color color ) [pure virtual]
```

Implemented in RayLib.

13.38.3.9 DrawCube()

```
virtual void IGraphicsLib::DrawCube (   
    ZappyTypes::Vector3 position,  
    float width,  
    float height,  
    float length,  
    ZappyTypes::Color color ) [pure virtual]
```

Implemented in RayLib.

13.38.3.10 DrawCylinder()

```
virtual void IGraphicsLib::DrawCylinder (   
    ZappyTypes::Vector3 position,  
    float radiusTop,  
    float radiusBottom,  
    float height,  
    int slices,  
    ZappyTypes::Color color ) [pure virtual]
```

Implemented in RayLib.

13.38.3.11 DrawGrid()

```
virtual void IGraphicsLib::DrawGrid (   
    int slices,  
    float spacing ) [pure virtual]
```

Implemented in RayLib.

13.38.3.12 DrawLine3D()

```
virtual void IGraphicsLib::DrawLine3D (
    ZappyTypes::Vector3 startPos,
    ZappyTypes::Vector3 endPos,
    ZappyTypes::Color color ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.13 DrawModel3D()

```
virtual void IGraphicsLib::DrawModel3D (
    int modelId,
    ZappyTypes::Vector3 position,
    float scale,
    ZappyTypes::Color color ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.14 DrawModelEx()

```
virtual void IGraphicsLib::DrawModelEx (
    int modelId,
    ZappyTypes::Vector3 position,
    ZappyTypes::Vector3 rotationAxis,
    float rotationAngle,
    float scale ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.15 DrawPlane()

```
virtual void IGraphicsLib::DrawPlane (
    ZappyTypes::Vector3 centerPos,
    ZappyTypes::Vector2 size,
    ZappyTypes::Color color ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.16 DrawRectangle()

```
virtual void IGraphicsLib::DrawRectangle (
    int x,
    int y,
    int width,
    int height,
    ZappyTypes::Color color ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.17 DrawSphere()

```
virtual void IGraphicsLib::DrawSphere (
    ZappyTypes::Vector3 centerPos,
    float radius,
    ZappyTypes::Color color ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.18 DrawText()

```
virtual void IGraphicsLib::DrawText (
    const std::string & text,
    int x,
    int y,
    int size,
    ZappyTypes::Color color ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.19 DrawText3D()

```
virtual void IGraphicsLib::DrawText3D (
    const std::string & text,
    ZappyTypes::Vector3 position,
    float fontSize,
    float fontSpacing,
    float lineSpacing,
    bool backface,
    ZappyTypes::Color tint ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.20 DrawTexture2D()

```
virtual void IGraphicsLib::DrawTexture2D (
    int textureId,
    int x,
    int y ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.21 DrawTextWave3D()

```
virtual void IGraphicsLib::DrawTextWave3D (
    const std::string & text,
    ZappyTypes::Vector3 position,
    float fontSize,
    float fontSpacing,
    float lineSpacing,
    bool backface,
    float time,
    ZappyTypes::Color tint ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.22 EndCamera3D()

```
virtual void IGraphicsLib::EndCamera3D ( ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.23 EndDrawing()

```
virtual void IGraphicsLib::EndDrawing ( ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.24 GetMousePosition()

```
virtual ZappyTypes::Vector2 IGraphicsLib::GetMousePosition () [pure virtual]  
Implemented in RayLib.
```

13.38.3.25 GetMouseWheelMove()

```
virtual float IGraphicsLib::GetMouseWheelMove () [pure virtual]  
Implemented in RayLib.
```

13.38.3.26 GetMouseX()

```
virtual int IGraphicsLib::GetMouseX () [pure virtual]  
Implemented in RayLib.
```

13.38.3.27 GetMouseY()

```
virtual int IGraphicsLib::GetMouseY () [pure virtual]  
Implemented in RayLib.
```

13.38.3.28 InitWindow()

```
virtual void IGraphicsLib::InitWindow (  
    int width,  
    int height,  
    const std::string & title ) [pure virtual]  
Implemented in RayLib.
```

13.38.3.29 IsKeyDown()

```
virtual bool IGraphicsLib::IsKeyDown (  
    int key ) [pure virtual]  
Implemented in RayLib.
```

13.38.3.30 IsKeyPressed()

```
virtual bool IGraphicsLib::IsKeyPressed (  
    int key ) [pure virtual]  
Implemented in RayLib.
```

13.38.3.31 IsMouseButtonDown()

```
virtual bool IGraphicsLib::IsMouseButtonDown (  
    int button ) [pure virtual]  
Implemented in RayLib.
```

13.38.3.32 IsMouseButtonPressed()

```
virtual bool IGraphicsLib::IsMouseButtonPressed (  
    int button ) [pure virtual]  
Implemented in RayLib.
```

13.38.3.33 IsMouseButtonReleased()

```
virtual bool IGraphicsLib::IsMouseButtonReleased (
    int button ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.34 IsTextureReady()

```
virtual bool IGraphicsLib::IsTextureReady (
    int textureId ) const [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.35 LoadFont()

```
virtual void IGraphicsLib::LoadFont (
    const std::string & path ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.36 LoadModel3D()

```
virtual int IGraphicsLib::LoadModel3D (
    const std::string & path ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.37 LoadModelWithTexture()

```
virtual int IGraphicsLib::LoadModelWithTexture (
    const std::string & modelPath,
    const std::string & texturePath ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.38 LoadTexture2D()

```
virtual int IGraphicsLib::LoadTexture2D (
    const std::string & path ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.39 LoadTexture3D()

```
virtual void IGraphicsLib::LoadTexture3D (
    const std::string & path ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.40 MeasureText3D()

```
virtual ZappyTypes::Vector3 IGraphicsLib::MeasureText3D (
    const std::string & text,
    float fontSize,
    float fontSpacing,
    float lineSpacing ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.41 PlayMusic()

```
virtual void IGraphicsLib::PlayMusic (
    const std::string & path ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.42 PlaySound()

```
virtual void IGraphicsLib::PlaySound (
    const std::string & path ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.43 setFps()

```
virtual void IGraphicsLib::setFps (
    int fps ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.44 SetMusicVolume()

```
virtual void IGraphicsLib::SetMusicVolume (
    float volume ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.45 SetSoundVolume()

```
virtual void IGraphicsLib::SetSoundVolume (
    float volume ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.46 StopMusic()

```
virtual void IGraphicsLib::StopMusic ( ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.47 StopSound()

```
virtual void IGraphicsLib::StopSound ( ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.48 UnloadFont()

```
virtual void IGraphicsLib::UnloadFont ( ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.49 UnloadModel3D()

```
virtual void IGraphicsLib::UnloadModel3D (
    int modelId ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.50 UnloadTexture2D()

```
virtual void IGraphicsLib::UnloadTexture2D (
    int textureId ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.51 UnloadTexture3D()

```
virtual void IGraphicsLib::UnloadTexture3D ( ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.52 UpdateMusic()

```
virtual void IGraphicsLib::UpdateMusic ( ) [pure virtual]
```

Implemented in [RayLib](#).

13.38.3.53 WindowShouldClose()

```
virtual bool IGraphicsLib::WindowShouldClose ( ) [pure virtual]
```

Implemented in [RayLib](#).

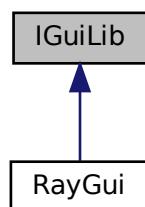
The documentation for this class was generated from the following file:

- /root/Desktop/Zappy/src/Shared/[IGraphicsLib.hpp](#)

13.39 IGuiLib Class Reference

```
#include <IGuiLib.hpp>
```

Inheritance diagram for IGuiLib:



Public Member Functions

- virtual [~IGuiLib](#) ()=default
- virtual void [DrawButton](#) (float x, float y, float width, float height, const std::string &text)=0
- virtual bool [ButtonPressed](#) (float x, float y, float width, float height, const std::string &text)=0
- virtual void [DrawLabel](#) (float x, float y, float width, float height, const std::string &text)=0
- virtual bool [DrawCheckBox](#) (float x, float y, float width, float height, const std::string &text, bool checked)=0
- virtual bool [DrawToggle](#) (float x, float y, float width, float height, const std::string &text, bool active)=0
- virtual int [DrawToggleGroup](#) (float x, float y, float width, float height, const std::string &text, int active)=0
- virtual int [DrawComboBox](#) (float x, float y, float width, float height, const std::string &text, int active)=0

- virtual int [DrawDropdownBox](#) (float x, float y, float width, float height, const std::string &text, int active, bool editMode)=0
- virtual bool [DrawTextBox](#) (float x, float y, float width, float height, std::shared_ptr< std::string > text, int maxLength, bool editMode)=0
- virtual bool [DrawValueBox](#) (float x, float y, float width, float height, std::shared_ptr< int > value, int minValue, int maxValue, bool editMode)=0
- virtual bool [DrawSpinner](#) (float x, float y, float width, float height, std::shared_ptr< int > value, int minValue, int maxValue, bool editMode)=0
- virtual float [DrawSlider](#) (float x, float y, float width, float height, const std::string &textLeft, const std::string &textRight, float value, float minValue, float maxValue)=0
- virtual float [DrawProgressBar](#) (float x, float y, float width, float height, const std::string &textLeft, const std::string &textRight, float value, float minValue, float maxValue)=0
- virtual void [DrawStatusBar](#) (float x, float y, float width, float height, const std::string &text)=0
- virtual bool [DrawWindowBox](#) (float x, float y, float width, float height, const std::string &title)=0
- virtual void [DrawGroupBox](#) (float x, float y, float width, float height, const std::string &text)=0
- virtual void [DrawPanel](#) (float x, float y, float width, float height)=0
- virtual ZappyTypes::Rectangle [DrawScrollPane](#) (float x, float y, float width, float height, const std::string &text, ZappyTypes::Rectangle content, std::shared_ptr< ZappyTypes::Vector2 > scroll)=0
- virtual int [DrawTabBar](#) (float x, float y, float width, float height, const std::vector< std::string > &tabs, int active)=0
- virtual int [DrawListView](#) (float x, float y, float width, float height, const std::vector< std::string > &items, int active)=0
- virtual ZappyTypes::Color [DrawColorPicker](#) (float x, float y, float width, float height, ZappyTypes::Color color)=0
- virtual int [DrawMessageBox](#) (float x, float y, float width, float height, const std::string &title, const std::string &message, const std::string &buttons)=0
- virtual int [DrawTextInputBox](#) (float x, float y, float width, float height, const std::string &title, const std::string &message, std::shared_ptr< std::string > text, const std::string &buttons)=0
- virtual std::string [IconText](#) (int iconId, const std::string &text)=0
- virtual void [LoadStyle](#) (const std::string &file)=0
- virtual void [SetStyle](#) (int control, int property, int value)=0
- virtual int [GetStyle](#) (int control, int property)=0

13.39.1 Detailed Description

Definition at line 14 of file IGuiLib.hpp.

13.39.2 Constructor & Destructor Documentation

13.39.2.1 ~IGuiLib()

```
virtual IGuiLib::~IGuiLib() [virtual], [default]
```

13.39.3 Member Function Documentation

13.39.3.1 ButtonPressed()

```
virtual bool IGuiLib::ButtonPressed (
    float x,
    float y,
    float width,
    float height,
    const std::string & text ) [pure virtual]
```

Implemented in [RayGui](#).

13.39.3.2 DrawButton()

```
virtual void IGuiLib::DrawButton (
    float x,
    float y,
    float width,
    float height,
    const std::string & text ) [pure virtual]
```

Implemented in [RayGui](#).

13.39.3.3 DrawCheckBox()

```
virtual bool IGuiLib::DrawCheckBox (
    float x,
    float y,
    float width,
    float height,
    const std::string & text,
    bool checked ) [pure virtual]
```

Implemented in [RayGui](#).

13.39.3.4 DrawColorPicker()

```
virtual ZappyTypes::Color IGuiLib::DrawColorPicker (
    float x,
    float y,
    float width,
    float height,
    ZappyTypes::Color color ) [pure virtual]
```

Implemented in [RayGui](#).

13.39.3.5 DrawComboBox()

```
virtual int IGuiLib::DrawComboBox (
    float x,
    float y,
    float width,
    float height,
    const std::string & text,
    int active ) [pure virtual]
```

Implemented in [RayGui](#).

13.39.3.6 DrawDropdownBox()

```
virtual int IGuiLib::DrawDropdownBox (
    float x,
    float y,
    float width,
    float height,
    const std::string & text,
    int active,
    bool editMode ) [pure virtual]
```

Implemented in [RayGui](#).

13.39.3.7 DrawGroupBox()

```
virtual void IGuiLib::DrawGroupBox (
    float x,
    float y,
    float width,
    float height,
    const std::string & text ) [pure virtual]
```

Implemented in [RayGui](#).

13.39.3.8 DrawLabel()

```
virtual void IGuiLib::DrawLabel (
    float x,
    float y,
    float width,
    float height,
    const std::string & text ) [pure virtual]
```

Implemented in [RayGui](#).

13.39.3.9 DrawListView()

```
virtual int IGuiLib::DrawListView (
    float x,
    float y,
    float width,
    float height,
    const std::vector< std::string > & items,
    int active ) [pure virtual]
```

Implemented in [RayGui](#).

13.39.3.10 DrawMessageBox()

```
virtual int IGuiLib::DrawMessageBox (
    float x,
    float y,
    float width,
    float height,
    const std::string & title,
    const std::string & message,
    const std::string & buttons ) [pure virtual]
```

Implemented in [RayGui](#).

13.39.3.11 DrawPanel()

```
virtual void IGuiLib::DrawPanel (
    float x,
    float y,
    float width,
    float height ) [pure virtual]
```

Implemented in [RayGui](#).

13.39.3.12 DrawProgressBar()

```
virtual float IGuiLib::DrawProgressBar (
```

```

    float x,
    float y,
    float width,
    float height,
    const std::string & textLeft,
    const std::string & textRight,
    float value,
    float minValue,
    float maxValue ) [pure virtual]

```

Implemented in [RayGui](#).

13.39.3.13 DrawScrollPane()

```

virtual ZappyTypes::Rectangle IGuiLib::DrawScrollPane (
    float x,
    float y,
    float width,
    float height,
    const std::string & text,
    ZappyTypes::Rectangle content,
    std::shared_ptr< ZappyTypes::Vector2 > scroll ) [pure virtual]

```

Implemented in [RayGui](#).

13.39.3.14 DrawSlider()

```

virtual float IGuiLib::DrawSlider (
    float x,
    float y,
    float width,
    float height,
    const std::string & textLeft,
    const std::string & textRight,
    float value,
    float minValue,
    float maxValue ) [pure virtual]

```

Implemented in [RayGui](#).

13.39.3.15 DrawSpinner()

```

virtual bool IGuiLib::DrawSpinner (
    float x,
    float y,
    float width,
    float height,
    std::shared_ptr< int > value,
    int minValue,
    int maxValue,
    bool editMode ) [pure virtual]

```

Implemented in [RayGui](#).

13.39.3.16 DrawStatusBar()

```

virtual void IGuiLib::DrawStatusBar (
    float x,
    float y,

```

```
    float width,
    float height,
    const std::string & text ) [pure virtual]
```

Implemented in [RayGui](#).

13.39.3.17 DrawTabBar()

```
virtual int IGuiLib::DrawTabBar (
    float x,
    float y,
    float width,
    float height,
    const std::vector< std::string > & tabs,
    int active ) [pure virtual]
```

Implemented in [RayGui](#).

13.39.3.18 DrawTextBox()

```
virtual bool IGuiLib::DrawTextBox (
    float x,
    float y,
    float width,
    float height,
    std::shared_ptr< std::string > text,
    int maxLength,
    bool editMode ) [pure virtual]
```

Implemented in [RayGui](#).

13.39.3.19 DrawTextInputBox()

```
virtual int IGuiLib::DrawTextInputBox (
    float x,
    float y,
    float width,
    float height,
    const std::string & title,
    const std::string & message,
    std::shared_ptr< std::string > text,
    const std::string & buttons ) [pure virtual]
```

Implemented in [RayGui](#).

13.39.3.20 DrawToggle()

```
virtual bool IGuiLib::DrawToggle (
    float x,
    float y,
    float width,
    float height,
    const std::string & text,
    bool active ) [pure virtual]
```

Implemented in [RayGui](#).

13.39.3.21 DrawToggleGroup()

```
virtual int IGuiLib::DrawToggleGroup (
    float x,
    float y,
    float width,
    float height,
    const std::string & text,
    int active ) [pure virtual]
```

Implemented in [RayGui](#).

13.39.3.22 DrawValueBox()

```
virtual bool IGuiLib::DrawValueBox (
    float x,
    float y,
    float width,
    float height,
    std::shared_ptr< int > value,
    int minValue,
    int maxValue,
    bool editMode ) [pure virtual]
```

Implemented in [RayGui](#).

13.39.3.23 DrawWindowBox()

```
virtual bool IGuiLib::DrawWindowBox (
    float x,
    float y,
    float width,
    float height,
    const std::string & title ) [pure virtual]
```

Implemented in [RayGui](#).

13.39.3.24 GetStyle()

```
virtual int IGuiLib::GetStyle (
    int control,
    int property ) [pure virtual]
```

Implemented in [RayGui](#).

13.39.3.25 IconText()

```
virtual std::string IGuiLib::IconText (
    int iconId,
    const std::string & text ) [pure virtual]
```

Implemented in [RayGui](#).

13.39.3.26 LoadStyle()

```
virtual void IGuiLib::LoadStyle (
    const std::string & file ) [pure virtual]
```

Implemented in [RayGui](#).

13.39.3.27 SetStyle()

```
virtual void IGuiLib::SetStyle (
    int control,
    int property,
    int value ) [pure virtual]
```

Implemented in [RayGui](#).

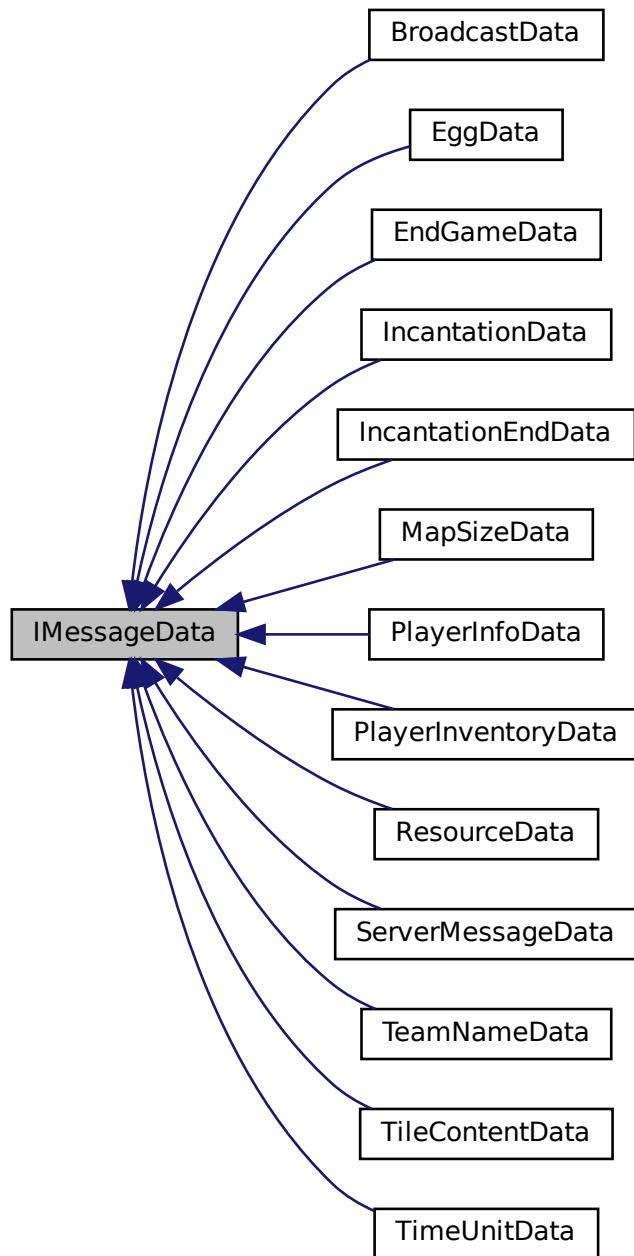
The documentation for this class was generated from the following file:

- /root/Desktop/Zappy/src/Shared/[IGuiLib.hpp](#)

13.40 IMESSAGEData Class Reference

```
#include <IMessageData.hpp>
```

Inheritance diagram for IMESSAGEData:



Public Member Functions

- virtual `~IMESSAGEData ()=default`
- virtual `MessageType getType () const =0`

13.40.1 Detailed Description

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

13.40.2 Constructor & Destructor Documentation

13.40.2.1 ~IMessageData()

```
virtual IMessageData::~IMessageData ( ) [virtual], [default]
```

13.40.3 Member Function Documentation

13.40.3.1 getType()

```
virtual MessageType IMessageData::getType ( ) const [pure virtual]
```

Implemented in [TimeUnitData](#), [TileContentData](#), [TeamNameData](#), [ServerMessageData](#), [ResourceData](#), [PlayerInventoryData](#), [PlayerInfoData](#), [MapSizeData](#), [IncantationEndData](#), [IncantationData](#), [EndGameData](#), [EggData](#), and [BroadcastData](#).

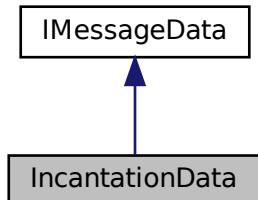
The documentation for this class was generated from the following file:

- /root/Desktop/Zappy/src/GUI/network/protocol/messageData/[IMessageData.hpp](#)

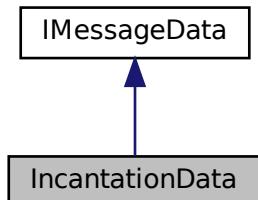
13.41 IncantationData Class Reference

```
#include <IncantationData.hpp>
```

Inheritance diagram for IncantationData:



Collaboration diagram for IncantationData:



Public Member Functions

- `IncantationData` (int x, int y, int level, const std::vector< int > &playerIds)
- `MessageType getType` () const override
- int `getX` () const
- int `getY` () const
- int `getLevel` () const
- const std::vector< int > & `getPlayerIds` () const
- void `setX` (int value)
- void `setY` (int value)
- void `setLevel` (int value)
- void `setPlayerIds` (const std::vector< int > &value)
- void `addPlayerId` (int id)

13.41.1 Detailed Description

Definition at line 14 of file IncantationData.hpp.

13.41.2 Constructor & Destructor Documentation

13.41.2.1 IncantationData()

```
IncantationData::IncantationData (
    int x,
    int y,
    int level,
    const std::vector< int > & playerIds ) [inline]
Definition at line 16 of file IncantationData.hpp.
17 : _x(x), _y(y), _level(level), _playerIds(playerIds) {}
```

13.41.3 Member Function Documentation

13.41.3.1 addPlayerId()

```
void IncantationData::addPlayerId (
    int id ) [inline]
Definition at line 29 of file IncantationData.hpp.
29 { _playerIds.push_back(id); }
```

13.41.3.2 getLevel()

```
int IncantationData::getLevel ( ) const [inline]
Definition at line 22 of file IncantationData.hpp.
22 { return _level; }
```

13.41.3.3 getPlayerIds()

```
const std::vector<int>& IncantationData::getPlayerIds ( ) const [inline]
Definition at line 23 of file IncantationData.hpp.
23 { return _playerIds; }
```

13.41.3.4 getType()

```
MessageType IncantationData::getType () const [inline], [override], [virtual]
Implements IMessageData.  
Definition at line 18 of file IncantationData.hpp.  
18 { return MessageType::Incantation; }
```

13.41.3.5 getX()

```
int IncantationData::getX () const [inline]
Definition at line 20 of file IncantationData.hpp.  
20 { return _x; }
```

13.41.3.6 getY()

```
int IncantationData::getY () const [inline]
Definition at line 21 of file IncantationData.hpp.  
21 { return _y; }
```

13.41.3.7 setLevel()

```
void IncantationData::setLevel (
    int value ) [inline]
Definition at line 27 of file IncantationData.hpp.  
27 { _level = value; }
```

13.41.3.8 setPlayerIds()

```
void IncantationData::setPlayerIds (
    const std::vector< int > & value ) [inline]
Definition at line 28 of file IncantationData.hpp.  
28 { _playerIds = value; }
```

13.41.3.9 setX()

```
void IncantationData::setX (
    int value ) [inline]
Definition at line 25 of file IncantationData.hpp.  
25 { _x = value; }
```

13.41.3.10 setY()

```
void IncantationData::setY (
    int value ) [inline]
Definition at line 26 of file IncantationData.hpp.  
26 { _y = value; }
```

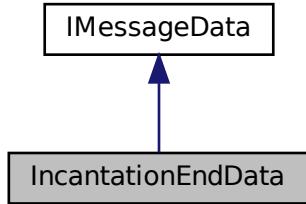
The documentation for this class was generated from the following file:

- /root/Desktop/Zappy/src/GUI/network/protocol/messageData/[IncantationData.hpp](#)

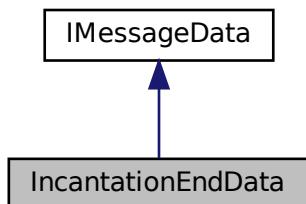
13.42 IncantationEndData Class Reference

```
#include <IncantationEndData.hpp>
```

Inheritance diagram for IncantationEndData:



Collaboration diagram for IncantationEndData:



Public Member Functions

- `IncantationEndData (int x, int y, bool success)`
- `MessageType getType () const override`
- `int getX () const`
- `int getY () const`
- `bool isSuccess () const`
- `void setX (int value)`
- `void setY (int value)`
- `void setSuccess (bool value)`

13.42.1 Detailed Description

Definition at line 13 of file IncantationEndData.hpp.

13.42.2 Constructor & Destructor Documentation

13.42.2.1 IncantationEndData()

```
IncantationEndData::IncantationEndData (
    int x,
```

```

    int y,
    bool success ) [inline]
Definition at line 15 of file IncantationEndData.hpp.
16 : _x(x), _y(y), _success(success) {}

```

13.42.3 Member Function Documentation

13.42.3.1 getType()

```

MessageType IncantationEndData::getType () const [inline], [override], [virtual]
Implements IMessageData.
Definition at line 18 of file IncantationEndData.hpp.
18 { return MessageType::IncantationEnd; }

```

13.42.3.2 getX()

```

int IncantationEndData::getX () const [inline]
Definition at line 20 of file IncantationEndData.hpp.
20 { return _x; }

```

13.42.3.3 getY()

```

int IncantationEndData::getY () const [inline]
Definition at line 21 of file IncantationEndData.hpp.
21 { return _y; }

```

13.42.3.4 isSuccess()

```

bool IncantationEndData::isSuccess () const [inline]
Definition at line 22 of file IncantationEndData.hpp.
22 { return _success; }

```

13.42.3.5 setSuccess()

```

void IncantationEndData::setSuccess (
    bool value ) [inline]
Definition at line 26 of file IncantationEndData.hpp.
26 { _success = value; }

```

13.42.3.6 setX()

```

void IncantationEndData::setX (
    int value ) [inline]
Definition at line 24 of file IncantationEndData.hpp.
24 { _x = value; }

```

13.42.3.7 setY()

```

void IncantationEndData::setY (
    int value ) [inline]
Definition at line 25 of file IncantationEndData.hpp.
25 { _y = value; }

```

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

- /root/Desktop/Zappy/src/GUI/network/protocol/messageData/[IncantationEndData.hpp](#)

13.43 raylibcpp::Input Class Reference

```
#include <Input.hpp>
```

Static Public Member Functions

- static bool `isKeyPressed` (int key)
- static bool `isMouseButtonPressed` (int button)
- static bool `isKeyDown` (int key)
- static bool `isMouseButtonDown` (int button)
- static bool `isMouseButtonReleased` (int button)
- static int `getMouseX` ()
- static int `getMouseY` ()
- static Vector2 `getMousePosition` ()
- static float `getMouseWheelMove` ()

13.43.1 Detailed Description

Definition at line 13 of file Input.hpp.

13.43.2 Member Function Documentation

13.43.2.1 `getMousePosition()`

```
Vector2 raylibcpp::Input::getMousePosition () [static]
```

Definition at line 40 of file Input.cpp.

```
40
41     return Get.mousePosition();
42 }
```

13.43.2.2 `getMouseWheelMove()`

```
float raylibcpp::Input::getMouseWheelMove () [static]
```

Definition at line 44 of file Input.cpp.

```
44
45     return Get.mousePosition();
46 }
```

13.43.2.3 `getMouseX()`

```
int raylibcpp::Input::getMouseX () [static]
```

Definition at line 32 of file Input.cpp.

```
32
33     return Get.mousePosition();
34 }
```

13.43.2.4 `getMouseY()`

```
int raylibcpp::Input::getMouseY () [static]
```

Definition at line 36 of file Input.cpp.

```
36
37     return Get.mousePosition();
38 }
```

13.43.2.5 isKeyDown()

```
bool raylibcpp::Input::isKeyDown (
    int key ) [static]
Definition at line 20 of file Input.cpp.
20
21     return IsKeyDown(key);
22 }
```

13.43.2.6 isKeyPressed()

```
bool raylibcpp::Input::isKeyPressed (
    int key ) [static]
Definition at line 12 of file Input.cpp.
12
13     return IsKeyPressed(key);
14 }
```

13.43.2.7 isMouseButtonDown()

```
bool raylibcpp::Input::isMouseButtonDown (
    int button ) [static]
Definition at line 24 of file Input.cpp.
24
25     return IsMouseButtonDown(button);
26 }
```

13.43.2.8 isMouseButtonPressed()

```
bool raylibcpp::Input::isMouseButtonPressed (
    int button ) [static]
Definition at line 16 of file Input.cpp.
16
17     return IsMouseButtonPressed(button);
18 }
```

13.43.2.9 isMouseButtonReleased()

```
bool raylibcpp::Input::isMouseButtonReleased (
    int button ) [static]
Definition at line 28 of file Input.cpp.
28
29     return IsMouseButtonReleased(button);
30 }
```

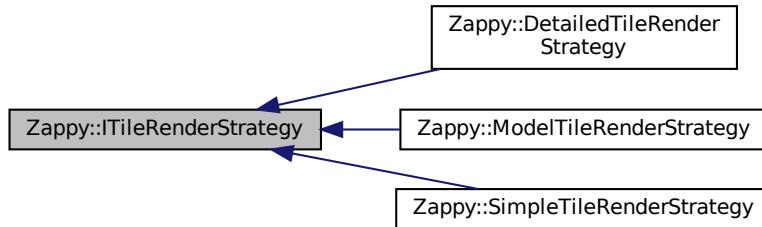
The documentation for this class was generated from the following files:

- /root/Desktop/Zappy/libs/RaylibCPP/input/[Input.hpp](#)
- /root/Desktop/Zappy/libs/RaylibCPP/input/[Input.cpp](#)

13.44 Zappy::ITileRenderStrategy Class Reference

```
#include <ITileRenderStrategy.hpp>
```

Inheritance diagram for Zappy::ITileRenderStrategy:



Public Member Functions

- virtual `~ITileRenderStrategy ()=default`
- virtual void `renderTile (const std::shared_ptr< IGraphicsLib > &graphicsLib, int x, int y, const ZappyTypes::Color &color, float tileSize, float spacing)=0`

13.44.1 Detailed Description

Definition at line 14 of file ITileRenderStrategy.hpp.

13.44.2 Constructor & Destructor Documentation

13.44.2.1 ~ITileRenderStrategy()

```
virtual Zappy::ITileRenderStrategy::~ITileRenderStrategy ( ) [virtual], [default]
```

13.44.3 Member Function Documentation

13.44.3.1 renderTile()

```
virtual void Zappy::ITileRenderStrategy::renderTile (
    const std::shared_ptr< IGraphicsLib > & graphicsLib,
    int x,
    int y,
    const ZappyTypes::Color & color,
    float tileSize,
    float spacing ) [pure virtual]
```

Implemented in [Zappy::SimpleTileRenderStrategy](#), [Zappy::ModelTileRenderStrategy](#), and [Zappy::DetailedTileRenderStrategy](#).
The documentation for this class was generated from the following file:

- /root/Desktop/Zappy/src/GUI/renderer/strategies/ITileRenderStrategy.hpp

13.45 RayGUICPP::Label Class Reference

```
#include <RayGuiEncap.hpp>
```

Static Public Member Functions

- static void [Draw](#) (const Rectangle &bounds, const std::string &text)

13.45.1 Detailed Description

Definition at line 41 of file RayGuiEncap.hpp.

13.45.2 Member Function Documentation

13.45.2.1 Draw()

```
static void RayGUICPP::Label::Draw (
    const Rectangle & bounds,
    const std::string & text) [inline], [static]
```

Definition at line 43 of file RayGuiEncap.hpp.

```
43
44     GuiLabel(bounds, text.c_str());
45 }
```

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

- /root/Desktop/Zappy/libs/RayGUICPP/include/[RayGuiEncap.hpp](#)

13.46 LibraryManager Class Reference

```
#include <LibraryManager.hpp>
```

Public Member Functions

- bool [loadGraphicsLib](#) (const std::string &libPath, const std::string &creatorFunction="createGraphicsLib")
Charge une bibliothèque graphique.
- bool [loadGuiLib](#) (const std::string &libPath, const std::string &creatorFunction="createGuiLib")
Charge une bibliothèque GUI.
- std::shared_ptr<[IGraphicsLib](#)> [getGraphicsLibPtr](#) ()
Obtient la bibliothèque graphique chargée comme shared_ptr.
- std::shared_ptr<[IGuiLib](#)> [getGuiLibPtr](#) ()
Obtient la bibliothèque GUI chargée comme shared_ptr.
- [IGraphicsLib](#) & [getGraphicsLib](#) ()
Obtient la bibliothèque graphique chargée (pour rétrocompatibilité)
- [IGuiLib](#) & [getGuiLib](#) ()
Obtient la bibliothèque GUI chargée (pour rétrocompatibilité)
- const std::string & [getLastErr](#) () const
Obtient la dernière erreur générée.

Static Public Member Functions

- static [LibraryManager](#) & [getInstance](#) ()
Obtient l'instance singleton du gestionnaire de bibliothèques.

13.46.1 Detailed Description

Definition at line 17 of file LibraryManager.hpp.

13.46.2 Member Function Documentation

13.46.2.1 getGraphicsLib()

`IGraphicsLib& LibraryManager::getGraphicsLib () [inline]`
 Obtient la bibliothèque graphique chargée (pour rétrocompatibilité)

Returns

Référence vers la bibliothèque graphique

Definition at line 90 of file LibraryManager.hpp.

```
90
91     if (!_graphicsLib) {
92         throw std::runtime_error("Bibliothèque graphique non chargée");
93     }
94     return *_graphicsLib;
95 }
```

13.46.2.2 getGraphicsLibPtr()

`std::shared_ptr<IGraphicsLib> LibraryManager::getGraphicsLibPtr () [inline]`
 Obtient la bibliothèque graphique chargée comme shared_ptr.

Returns

shared_ptr vers la bibliothèque graphique

Definition at line 66 of file LibraryManager.hpp.

```
66
67     if (!_graphicsLib) {
68         throw std::runtime_error("Bibliothèque graphique non chargée");
69     }
70     // Crée un shared_ptr qui ne détruit pas l'objet (il est déjà géré par _graphicsLib)
71     return std::shared_ptr<IGraphicsLib>(_graphicsLib.get(), [](IGraphicsLib*) {});
72 }
```

13.46.2.3 getGuiLib()

`IGuiLib& LibraryManager::getGuiLib () [inline]`
 Obtient la bibliothèque GUI chargée (pour rétrocompatibilité)

Returns

Référence vers la bibliothèque GUI

Definition at line 101 of file LibraryManager.hpp.

```
101
102     if (!_guiLib) {
103         throw std::runtime_error("Bibliothèque GUI non chargée");
104     }
105     return *_guiLib;
106 }
```

13.46.2.4 getGuiLibPtr()

`std::shared_ptr<IGuiLib> LibraryManager::getGuiLibPtr () [inline]`
 Obtient la bibliothèque GUI chargée comme shared_ptr.

Returns

shared_ptr vers la bibliothèque GUI

Definition at line 78 of file LibraryManager.hpp.

```
78
79     if (!_guiLib) {
80         throw std::runtime_error("Bibliothèque GUI non chargée");
81     }
82     // Crée un shared_ptr qui ne détruit pas l'objet (il est déjà géré par _guiLib)
83     return std::shared_ptr<IGuiLib>(_guiLib.get(), [](IGuiLib*) {});
84 }
```

13.46.2.5 getInstance()

```
static LibraryManager& LibraryManager::getInstance ( ) [inline], [static]
Obtient l'instance singleton du gestionnaire de bibliothèques.
```

Returns

Instance du [LibraryManager](#)

Definition at line 23 of file LibraryManager.hpp.

```
23
24     static LibraryManager instance;
25     return instance;
26 }
```

13.46.2.6 getLastError()

```
const std::string& LibraryManager::getLast Error ( ) const [inline]
Obtient la dernière erreur générée.
```

Returns

[Message](#) d'erreur

Definition at line 112 of file LibraryManager.hpp.

```
112
113     return _lastError;
114 }
```

13.46.2.7 loadGraphicsLib()

```
bool LibraryManager::loadGraphicsLib (
    const std::string & libPath,
    const std::string & creatorFunction = "createGraphicsLib" ) [inline]
```

Charge une bibliothèque graphique.

Parameters

<i>libPath</i>	Chemin vers la bibliothèque graphique
<i>creatorFunction</i>	Nom de la fonction de création (par défaut "createGraphicsLib")

Returns

true si chargée avec succès, false sinon

Definition at line 34 of file LibraryManager.hpp.

```
34
35     {
36         try {
37             _graphicsLoader = std::make_unique<DLLoader<IGraphicsLib>>(libPath);
38             _graphicsLib = _graphicsLoader->getInstance(creatorFunction);
39             return true;
40         } catch (const std::exception& e) {
41             _lastError = e.what();
42             return false;
43         }
44     }
```

13.46.2.8 loadGuiLib()

```
bool LibraryManager::loadGuiLib (
    const std::string & libPath,
    const std::string & creatorFunction = "createGuiLib" ) [inline]
```

Charge une bibliothèque GUI.

Parameters

<i>libPath</i>	Chemin vers la bibliothèque GUI
<i>creatorFunction</i>	Nom de la fonction de création (par défaut "createGuiLib")

Returns

true si chargée avec succès, false sinon

Definition at line 51 of file LibraryManager.hpp.

```

51
52     try {
53         _guiLoader = std::make_unique<DLLoader<IGuiLib>>(libPath);
54         _guiLib = _guiLoader->getInstance(creatorFunction);
55         return true;
56     } catch (const std::exception& e) {
57         _lastError = e.what();
58         return false;
59     }
60 }
```

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

- /root/Desktop/Zappy/src/Shared/LibraryManager.hpp

13.47 RayGUICPP::ListView Class Reference

```
#include <RayGuiEncap.hpp>
```

Static Public Member Functions

- static int [Draw](#) (const Rectangle &bounds, const std::vector< std::string > &items, int active)

13.47.1 Detailed Description

Definition at line 203 of file RayGuiEncap.hpp.

13.47.2 Member Function Documentation

13.47.2.1 Draw()

```
static int RayGUICPP::ListView::Draw (
    const Rectangle & bounds,
    const std::vector< std::string > & items,
    int active ) [inline], [static]
```

Definition at line 205 of file RayGuiEncap.hpp.

```

205
206     std::string joined;
207     for (size_t i = 0; i < items.size(); ++i) {
208         joined += items[i];
209         if (i + 1 < items.size()) joined += ";";
210     }
211     int value = active;
212     int scrollIndex = 0;
213     GuiListView(bounds, joined.c_str(), &scrollIndex, &value);
214     return value;
215 }
```

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

- /root/Desktop/Zappy/libs/RayGUICPP/include/RayGuiEncap.hpp

13.48 logger.Logger Class Reference

Public Member Functions

- def `__init__` (self, str fileName="AI.log", str `message_prefix`= "")
- def `log` (self, str message, LogLevel level)
- def `debug` (self, str message)
- def `info` (self, str message)
- def `warning` (self, str message)
- def `error` (self, str message)
- def `critical` (self, str message)

Data Fields

- `file`
- `message_prefix`
- `file_handle`

13.48.1 Detailed Description

Definition at line 11 of file logger.py.

13.48.2 Constructor & Destructor Documentation

13.48.2.1 `__init__()`

```
def logger.Logger.__init__ (
    self,
    str fileName = "AI.log",
    str message_prefix = "" )
```

Definition at line 12 of file logger.py.

```
12     def __init__(self, fileName: str = "AI.log", message_prefix: str = ""):
13         self.file = fileName
14         self.message_prefix = message_prefix
15         self.file_handle = open(fileName, "a")
```

13.48.3 Member Function Documentation

13.48.3.1 `critical()`

```
def logger.Logger.critical (
    self,
    str message )
```

Definition at line 35 of file logger.py.

```
35     def critical(self, message: str):
36         self.log(message, LogLevel.CRITICAL)
```

13.48.3.2 `debug()`

```
def logger.Logger.debug (
    self,
    str message )
```

Definition at line 23 of file logger.py.

```
23     def debug(self, message: str):
24         self.log(message, LogLevel.DEBUG)
25
```

13.48.3.3 error()

```
def logger.Logger.error (
    self,
    str message )
Definition at line 32 of file logger.py.
32     def error(self, message: str):
33         self.log(message, LogLevel.ERROR)
34
```

13.48.3.4 info()

```
def logger.Logger.info (
    self,
    str message )
Definition at line 26 of file logger.py.
26     def info(self, message: str):
27         self.log(message, LogLevel.INFO)
28
```

13.48.3.5 log()

```
def logger.Logger.log (
    self,
    str message,
    LogLevel level )
Definition at line 17 of file logger.py.
17     def log(self, message: str, level: LogLevel):
18         timestamp = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
19         level_name = level.name
20         self.file_handle.write(f"[{timestamp}] {level_name}: {self.message_prefix} {message}\n")
21         self.file_handle.flush()
22
```

13.48.3.6 warning()

```
def logger.Logger.warning (
    self,
    str message )
Definition at line 29 of file logger.py.
29     def warning(self, message: str):
30         self.log(message, LogLevel.WARNING)
31
```

13.48.4 Field Documentation

13.48.4.1 file

`logger.Logger.file`
 Definition at line 13 of file logger.py.

13.48.4.2 file_handle

`logger.Logger.file_handle`
 Definition at line 15 of file logger.py.

13.48.4.3 message_prefix

`logger.Logger.message_prefix`

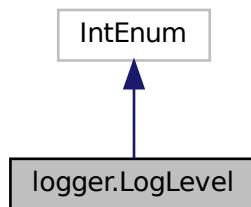
Definition at line 14 of file `logger.py`.

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

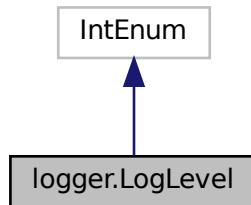
- /root/Desktop/Zappy/src/AI/logger/[logger.py](#)

13.49 logger.LogLevel Class Reference

Inheritance diagram for logger.LogLevel:



Collaboration diagram for logger.LogLevel:



Static Public Attributes

- int `DEBUG` = 10
- int `INFO` = 20
- int `WARNING` = 30
- int `ERROR` = 40
- int `CRITICAL` = 50

13.49.1 Detailed Description

Definition at line 4 of file `logger.py`.

13.49.2 Field Documentation

13.49.2.1 CRITICAL

```
int logger.LogLevel.CRITICAL = 50 [static]
Definition at line 9 of file logger.py.
```

13.49.2.2 DEBUG

```
int logger.LogLevel.DEBUG = 10 [static]
Definition at line 5 of file logger.py.
```

13.49.2.3 ERROR

```
int logger.LogLevel.ERROR = 40 [static]
Definition at line 8 of file logger.py.
```

13.49.2.4 INFO

```
int logger.LogLevel.INFO = 20 [static]
Definition at line 6 of file logger.py.
```

13.49.2.5 WARNING

```
int logger.LogLevel.WARNING = 30 [static]
Definition at line 7 of file logger.py.
```

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

- /root/Desktop/Zappy/src/AI/logger/logger.py

13.50 Zappy::MapRenderer Class Reference

Renderer de carte responsable de l'affichage de la grille. Implémente le pattern Bridge pour séparer l'interface du rendu.

```
#include <MapRenderer.hpp>
```

Public Member Functions

- **MapRenderer** (const std::shared_ptr<[IGraphicsLib](#)> &graphicsLib, const std::shared_ptr<[GraphicalContext](#)> &context, const std::shared_ptr<[ModelManagerAdapter](#)> &modelManagerAdapter)

Initialise le renderer de carte.
- void **initialize** ()

Initialise le renderer de carte.
- void **render** ()

Rend la carte complète.
- void **setTileRenderStrategy** (std::shared_ptr<[ITileRenderStrategy](#)> strategy)

Définit la stratégie de rendu pour toutes les tuiles.
- void **setResourceRenderStrategy** (int resourceType, std::shared_ptr<[ITileRenderStrategy](#)> strategy)

Définit une stratégie de rendu spécifique pour un type de ressource.
- void **setTileSize** (float size)

Modifie la taille des tuiles.
- void **setTileSpacing** (float spacing)

Modifie l'espacement entre les tuiles.
- void **setResourceColor** (int resourceType, const [ZappyTypes::Color](#) &color)

Définit la couleur pour un type de ressource spécifique.
- void **setZoomLevel** (float zoom)

Définit le niveau de zoom actuel.

- void **setDetailThreshold** (float threshold)

Définit le seuil de zoom pour afficher les détails.

13.50.1 Detailed Description

Renderer de carte responsable de l'affichage de la grille. Implémente le pattern Bridge pour séparer l'interface du rendu.

Definition at line 28 of file MapRenderer.hpp.

13.50.2 Constructor & Destructor Documentation

13.50.2.1 MapRenderer()

```
Zappy::MapRenderer::MapRenderer (
    const std::shared_ptr< IGraphicsLib > & graphicsLib,
    const std::shared_ptr< GraphicalContext > & context,
    const std::shared_ptr< ModelManagerAdapter > & modelManagerAdapter )
```

Definition at line 19 of file MapRenderer.cpp.

```
22     : graphicsLib(graphics),
23     context(ctx),
24     strategyFactory(modelManagerAdapter),
25     tileSize(1.0f),
26     tileSpacing(0.1f),
27     zoomLevel(1.0f),
28     detailThreshold(2.0f) {
29     tileRenderStrategy = strategyFactory.createSimpleTileStrategy(ctx);
30     detailedTileStrategy = strategyFactory.createDetailedTileStrategy(ctx);
31     std::shared_ptr<MapRenderer> selfPtr(this, [] (MapRenderer*) {});
32     context->addObserver(std::make_shared<MapRendererObserver>(selfPtr));
33 }
```

13.50.3 Member Function Documentation

13.50.3.1 initialize()

```
void Zappy::MapRenderer::initialize ( )
```

Initialise le renderer de carte.

Definition at line 35 of file MapRenderer.cpp.

```
35     {
36     resourceColors[static_cast<int>(ResourceType::FOOD)] = {255, 255, 150, 255};
37     resourceColors[static_cast<int>(ResourceType::LINEMATE)] = {100, 255, 100, 255};
38     resourceColors[static_cast<int>(ResourceType::DERAUMERE)] = {100, 100, 255, 255};
39     resourceColors[static_cast<int>(ResourceType::SIBUR)] = {255, 100, 100, 255};
40     resourceColors[static_cast<int>(ResourceType::MENDIANE)] = {200, 100, 200, 255};
41     resourceColors[static_cast<int>(ResourceType::PHIRAS)] = {255, 200, 100, 255};
42     resourceColors[static_cast<int>(ResourceType::THYSTAME)] = {100, 200, 255, 255};
43     resourceColors[-1] = {150, 150, 150, 255};
44 }
```

13.50.3.2 render()

```
void Zappy::MapRenderer::render ( )
```

Rend la carte complète.

Definition at line 46 of file MapRenderer.cpp.

```
46     {
47     if (!context || !graphicsLib)
48         return;
49     int mapWidth = context->getMapWidth();
50     int mapHeight = context->getMapHeight();
51     if (mapWidth <= 0 || mapHeight <= 0)
52         return;
53     for (int y = 0; y < mapHeight; ++y) {
54         for (int x = 0; x < mapWidth; ++x) {
```

```

55     ResourceType dominantResource = context->getDominantResourceType(x, y);
56     int resourceIndex = static_cast<int>(dominantResource);
57     renderTile(x, y, resourceIndex);
58   }
59 }
60 graphicsLib->DrawGrid(std::max(mapWidth, mapHeight), tileSize + tileSpacing);
61 }
```

13.50.3.3 setDetailThreshold()

```
void Zappy::MapRenderer::setDetailThreshold (
    float threshold )
```

Définit le seuil de zoom pour afficher les détails.

Parameters

<i>threshold</i>	Le seuil (typiquement entre 1.5 et 3.0)
------------------	---

Definition at line 89 of file MapRenderer.cpp.

```

89
90     detailThreshold = threshold;
91 }
```

13.50.3.4 setResourceColor()

```
void Zappy::MapRenderer::setResourceColor (
    int resourceType,
    const ZappyTypes::Color & color )
```

Définit la couleur pour un type de ressource spécifique.

Parameters

<i>resourceType</i>	Le type de ressource
<i>color</i>	La couleur à utiliser

Definition at line 81 of file MapRenderer.cpp.

```

81
82     resourceColors[resourceType] = color;
83 }
```

13.50.3.5 setResourceRenderStrategy()

```
void Zappy::MapRenderer::setResourceRenderStrategy (
    int resourceType,
    std::shared_ptr< ITileRenderStrategy > strategy )
```

Définit une stratégie de rendu spécifique pour un type de ressource.

Parameters

<i>resourceType</i>	Le type de ressource
<i>strategy</i>	La stratégie à utiliser

Definition at line 68 of file MapRenderer.cpp.

```

68
69     {
70       if (strategy)
71         resourceRenderStrategies[resourceType] = strategy;
72 }
```

13.50.3.6 setTileRenderStrategy()

```
void Zappy::MapRenderer::setTileRenderStrategy (
    std::shared_ptr< ITileRenderStrategy > strategy )
```

Définit la stratégie de rendu pour toutes les tuiles.

Parameters

<i>strategy</i>	La stratégie à utiliser
-----------------	-------------------------

Definition at line 63 of file MapRenderer.cpp.

```
63
64     if (strategy)
65         tileRenderStrategy = strategy;
66 }
```

13.50.3.7 setTileSize()

```
void Zappy::MapRenderer::setTileSize (
    float size )
```

Modifie la taille des tuiles.

Parameters

<i>size</i>	La nouvelle taille
-------------	--------------------

Definition at line 73 of file MapRenderer.cpp.

```
73
74     tileSize = std::max(0.1f, size);
75 }
```

13.50.3.8 setTileSpacing()

```
void Zappy::MapRenderer::setTileSpacing (
    float spacing )
```

Modifie l'espacement entre les tuiles.

Parameters

<i>spacing</i>	Le nouvel espacement
----------------	----------------------

Definition at line 77 of file MapRenderer.cpp.

```
77
78     tileSpacing = std::max(0.0f, spacing);
79 }
```

13.50.3.9 setZoomLevel()

```
void Zappy::MapRenderer::setZoomLevel (
    float zoom )
```

Définit le niveau de zoom actuel.

Parameters

<i>zoom</i>	Le niveau de zoom (1.0 = normal)
-------------	----------------------------------

Definition at line 85 of file MapRenderer.cpp.

```
85
86     zoomLevel = std::max(0.1f, zoom);
87 }
```

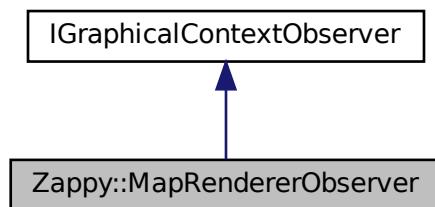
The documentation for this class was generated from the following files:

- /root/Desktop/Zappy/src/GUI/renderer/[MapRenderer.hpp](#)
- /root/Desktop/Zappy/src/GUI/renderer/[MapRenderer.cpp](#)

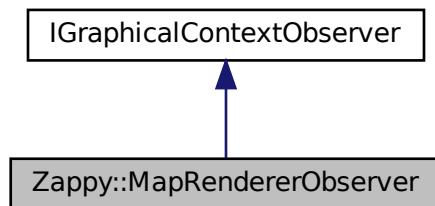
13.51 Zappy::MapRendererObserver Class Reference

```
#include <MapRendererObserver.hpp>
```

Inheritance diagram for Zappy::MapRendererObserver:



Collaboration diagram for Zappy::MapRendererObserver:



Public Member Functions

- [MapRendererObserver](#) (std::shared_ptr< [MapRenderer](#) > renderer)
- void [onMapSizeChanged](#) (int width, int height) override
- void [onTileChanged](#) (int x, int y, const [TileData](#) &tileData) override

13.51.1 Detailed Description

Definition at line 14 of file MapRendererObserver.hpp.

13.51.2 Constructor & Destructor Documentation

13.51.2.1 MapRendererObserver()

```
Zappy::MapRendererObserver::MapRendererObserver (
    std::shared_ptr< MapRenderer > renderer ) [explicit]
Definition at line 14 of file MapRendererObserver.cpp.
15     : renderer(renderer) {}
```

13.51.3 Member Function Documentation

13.51.3.1 onMapSizeChanged()

```
void Zappy::MapRendererObserver::onMapSizeChanged (
    int width,
    int height ) [override], [virtual]
```

Implements [IGraphicalContextObserver](#).

Definition at line 17 of file MapRendererObserver.cpp.

```
17
18     if (width > 20 || height > 20) {
19         float newSize = 10.0f / std::max(width, height);
20         renderer->setTileSize(newSize);
21         renderer->setTileSpacing(newSize * 0.1f);
22     }
23 }
```

13.51.3.2 onTileChanged()

```
void Zappy::MapRendererObserver::onTileChanged (
    int x,
    int y,
    const TileData & tileData ) [override], [virtual]
```

Implements [IGraphicalContextObserver](#).

Definition at line 25 of file MapRendererObserver.cpp.

```
25
26     //TODO(Sam) : maj avec les données du network
27 }
```

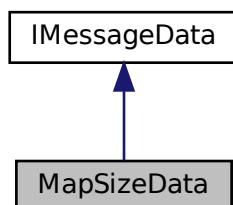
The documentation for this class was generated from the following files:

- [/root/Desktop/Zappy/src/GUI/renderer/MapRendererObserver.hpp](#)
- [/root/Desktop/Zappy/src/GUI/renderer/MapRendererObserver.cpp](#)

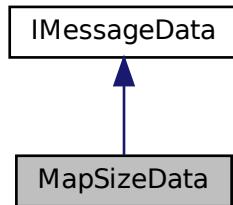
13.52 MapSizeData Class Reference

```
#include <MapSizeData.hpp>
```

Inheritance diagram for MapSizeData:



Collaboration diagram for MapSizeData:



Public Member Functions

- `MapSizeData (int width, int height)`
- `MessageType getType () const override`
- `int getWidth () const`
- `int getHeight () const`
- `void setWidth (int value)`
- `void setHeight (int value)`

13.52.1 Detailed Description

Definition at line 13 of file MapSizeData.hpp.

13.52.2 Constructor & Destructor Documentation

13.52.2.1 MapSizeData()

```

MapSizeData::MapSizeData (
    int width,
    int height ) [inline]
Definition at line 15 of file MapSizeData.hpp.
15 : _width(width), _height(height) {}
```

13.52.3 Member Function Documentation

13.52.3.1 getHeight()

```

int MapSizeData::getHeight ( ) const [inline]
Definition at line 19 of file MapSizeData.hpp.
19 { return _height; }
```

13.52.3.2 getType()

```

MessageType MapSizeData::getType ( ) const [inline], [override], [virtual]
Implements IMESSAGEData.
Definition at line 16 of file MapSizeData.hpp.
16 { return MessageType::MapSize; }
```

13.52.3.3 getWidth()

```
int MapSizeData::getWidth ( ) const [inline]
Definition at line 18 of file MapSizeData.hpp.
18 { return _width; }
```

13.52.3.4 setHeight()

```
void MapSizeData::setHeight (
    int value ) [inline]
Definition at line 22 of file MapSizeData.hpp.
22 { _height = value; }
```

13.52.3.5 setWidth()

```
void MapSizeData::setWidth (
    int value ) [inline]
Definition at line 21 of file MapSizeData.hpp.
21 { _width = value; }
```

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

- /root/Desktop/Zappy/src/GUI/network/protocol/messageData/MapSizeData.hpp

13.53 Message Class Reference

```
#include <Message.hpp>
```

Public Member Functions

- **Message ()**
Construit un message vide.
- **Message (const std::string &header, const std::string &data, std::shared_ptr< IMESSAGEData > structuredData=nullptr)**
Construit un message à partir d'un header, d'une data brute et d'une structure typée optionnelle.
- **~Message ()=default**
- **void setMessage (const std::string &data)**
Définit le message complet (header + data). Utilisé rarement.
- **void setHeader (const std::string &header)**
Définit le header (ex: "pnw").
- **void setData (const std::string &data)**
Définit la data brute (ex: "1 2 3 4 5 Team").
- **void setData (std::shared_ptr< IMESSAGEData > data)**
Définit la structure typée (IMESSAGEData).
- **const std::string & getMessage () const**
Retourne le message complet (header + data +).
- **const std::string & getHeader () const**
Retourne le header (ex: "pnw").
- **const std::string & getData () const**
Retourne la data brute (ex: "1 2 3 4 5 Team\n").
- **bool hasStructuredData () const**
Indique si une structure typée est présente.
- **std::shared_ptr< IMESSAGEData > getStructuredData () const**
Retourne la structure typée (IMESSAGEData) si présente.
- **void addIntParam (int value)**
- **void addStringParam (const std::string &value)**

13.53.1 Detailed Description

Definition at line 17 of file Message.hpp.

13.53.2 Constructor & Destructor Documentation

13.53.2.1 Message() [1/2]

```
Message::Message ( )
```

Construit un message vide.

Definition at line 17 of file Message.cpp.

```
17     {
18     _messageString = "";
19     _messageHeader = "";
20     _messageData = "";
21     _structuredData = nullptr;
22 }
```

13.53.2.2 Message() [2/2]

```
Message::Message (
    const std::string & header,
    const std::string & data,
    std::shared_ptr< IMESSAGEData > structuredData = nullptr )
```

Construit un message à partir d'un header, d'une data brute et d'une structure typée optionnelle.

Parameters

<i>header</i>	Le header du message (ex: "pnw")
<i>data</i>	La data brute (ex: "1 2 3 4 5 Team")
<i>structuredData</i>	Structure typée dérivée de IMESSAGEData (optionnelle)

Definition at line 24 of file Message.cpp.

```
24
25     {
26     _messageHeader = header;
27     _messageData = data;
28     if (_messageData.empty() || _messageData.back() != '\n')
29     _messageData += '\n';
30     _messageString = _messageHeader + " " + _messageData;
31 }
```

13.53.2.3 ~Message()

```
Message::~Message ( ) [default]
```

13.53.3 Member Function Documentation

13.53.3.1 addIntParam()

```
void Message::addIntParam (
    int value )
```

13.53.3.2 addStringParam()

```
void Message::addStringParam (
```

```
    const std::string & value )
```

13.53.3.3 getData()

const std::string & Message::getData () const
Retourne la data brute (ex: "1 2 3 4 5 Team\n").

Definition at line 60 of file Message.cpp.

```
60
61     return _messageData;
62 }
```

13.53.3.4 getHeader()

const std::string & Message::getHeader () const
Retourne le header (ex: "pnw").

Definition at line 56 of file Message.cpp.

```
56
57     return _messageHeader;
58 }
```

13.53.3.5 getMessage()

const std::string & Message::getMessage () const
Retourne le message complet (header + data +
).

Definition at line 52 of file Message.cpp.

```
52
53     return _messageString;
54 }
```

13.53.3.6 getStructuredData()

std::shared_ptr<IMessageData> Message::getStructuredData () const [inline]
Retourne la structure typée ([IMessageData](#)) si présente.

Definition at line 69 of file Message.hpp.

```
69 { return _structuredData; }
```

13.53.3.7 hasStructuredData()

bool Message::hasStructuredData () const
Indique si une structure typée est présente.

Definition at line 64 of file Message.cpp.

```
64
65     return _structuredData != nullptr;
66 }
```

13.53.3.8 setData() [1/2]

```
void Message::setData (
    const std::string & data )
```

Définit la data brute (ex: "1 2 3 4 5 Team").

Definition at line 41 of file Message.cpp.

```
41
42     _messageData = data;
43     if (_messageData.empty() || _messageData.back() != '\n')
44         _messageData += '\n';
45     _structuredData = nullptr;
46 }
```

13.53.3.9 setData() [2/2]

```
void Message::setData (
    std::shared_ptr< IMESSAGEData > data )
Définit la structure typée (IMESSAGEData).
```

Definition at line 48 of file Message.cpp.

```
48
49     _structuredData = data;
50 }
```

13.53.3.10 setHeader()

```
void Message::setHeader (
    const std::string & header )
Définit le header (ex: "pnw").
```

Definition at line 37 of file Message.cpp.

```
37
38     _messageHeader = header;
39 }
```

13.53.3.11 setMessage()

```
void Message::setMessage (
    const std::string & data )
Définit le message complet (header + data). Utilisé rarement.
```

Definition at line 33 of file Message.cpp.

```
33
34     _messageString = data;
35 }
```

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

- /root/Desktop/Zappy/src/GUI/network/protocol/[Message.hpp](#)
- /root/Desktop/Zappy/src/GUI/network/protocol/[Message.cpp](#)

13.54 RayGUICPP::MessageBox Class Reference

```
#include <RayGuiEncap.hpp>
```

Static Public Member Functions

- static int [Draw](#) (const Rectangle &bounds, const std::string &title, const std::string &message, const std::string &buttons)

13.54.1 Detailed Description

Definition at line 226 of file RayGuiEncap.hpp.

13.54.2 Member Function Documentation

13.54.2.1 Draw()

```
static int RayGUICPP::MessageBox::Draw (
    const Rectangle & bounds,
    const std::string & title,
    const std::string & message,
    const std::string & buttons ) [inline], [static]
```

Definition at line 228 of file RayGuiEncap.hpp.

```

228
229     {
230     }  

231     return GuiMessageBox(bounds, title.c_str(), message.c_str(), buttons.c_str());

```

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

- /root/Desktop/Zappy/libs/RayGUICPP/include/RayGuiEncap.hpp

13.55 MessageQueue Class Reference

```
#include <MessageQueue.hpp>
```

Public Member Functions

- [MessageQueue \(\)](#)
- [~MessageQueue \(\)=default](#)
- void [enqueue \(const std::string &message\)](#)
- std::string [dequeue \(\)](#)
- std::string [dequeueWait \(int timeoutMs=-1\)](#)
- bool [isEmpty \(\) const](#)
- size_t [size \(\) const](#)
- void [clear \(\)](#)

13.55.1 Detailed Description

Definition at line 17 of file MessageQueue.hpp.

13.55.2 Constructor & Destructor Documentation

13.55.2.1 MessageQueue()

```
MessageQueue::MessageQueue ( )  
Definition at line 13 of file MessageQueue.cpp.  
13 {  
14 }
```

13.55.2.2 ~MessageQueue()

```
MessageQueue::~MessageQueue ( ) [default]
```

13.55.3 Member Function Documentation

13.55.3.1 clear()

```
void MessageQueue::clear ( )  
Definition at line 58 of file MessageQueue.cpp.  
58 {
59     std::lock_guard<std::mutex> lock (_mutex);
60     while (!_queue.empty()) {
61         _queue.pop();
62     }
63 }
```

13.55.3.2 dequeue()

```
std::string MessageQueue::dequeue ( )
Definition at line 24 of file MessageQueue.cpp.
24
25     std::lock_guard<std::mutex> lock (_mutex);
26     if (_queue.empty())
27         return "";
28     std::string message = _queue.front();
29     _queue.pop();
30     return message;
31 }
```

13.55.3.3 dequeueWait()

```
std::string MessageQueue::dequeueWait (
    int timeoutMs = -1 )
Definition at line 33 of file MessageQueue.cpp.
33
34     std::unique_lock<std::mutex> lock (_mutex);
35     if (timeoutMs < 0) {
36         _cv.wait(lock, [this] { return !_queue.empty(); });
37     } else {
38         auto timeout = std::chrono::milliseconds(timeoutMs);
39         bool messageAvailable = _cv.wait_for(lock, timeout, [this] { return !_queue.empty(); });
40         if (!messageAvailable)
41             return "";
42     }
43     std::string message = _queue.front();
44     _queue.pop();
45     return message;
46 }
```

13.55.3.4 enqueue()

```
void MessageQueue::enqueue (
    const std::string & message )
Definition at line 16 of file MessageQueue.cpp.
16
17     {
18         std::lock_guard<std::mutex> lock (_mutex);
19         _queue.push(message);
20     }
21     _cv.notify_one();
22 }
```

13.55.3.5 isEmpty()

```
bool MessageQueue::isEmpty ( ) const
Definition at line 48 of file MessageQueue.cpp.
48
49     std::lock_guard<std::mutex> lock (_mutex);
50     return _queue.empty();
51 }
```

13.55.3.6 size()

```
size_t MessageQueue::size ( ) const
Definition at line 53 of file MessageQueue.cpp.
53
54     std::lock_guard<std::mutex> lock (_mutex);
55     return _queue.size();
56 }
```

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

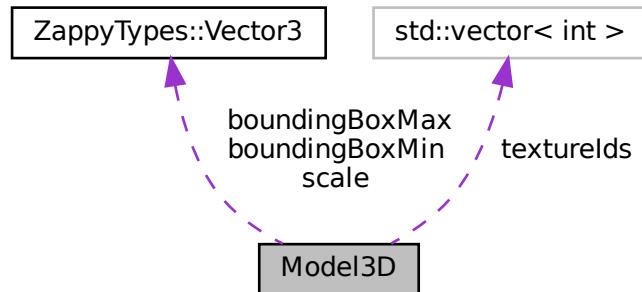
- /root/Desktop/Zappy/src/GUI/network/buffer/[MessageQueue.hpp](#)
- /root/Desktop/Zappy/src/GUI/network/buffer/[MessageQueue.cpp](#)

13.56 Model3D Struct Reference

Structure représentant un modèle 3D avec ses textures associées.

```
#include <ModelManager.hpp>
```

Collaboration diagram for Model3D:



Data Fields

- `int modelId`
- `std::vector< int > textureIds`
- `ZappyTypes::Vector3 scale`
- `ZappyTypes::Vector3 boundingBoxMin`
- `ZappyTypes::Vector3 boundingBoxMax`

13.56.1 Detailed Description

Structure représentant un modèle 3D avec ses textures associées.

Definition at line 24 of file ModelManager.hpp.

13.56.2 Field Documentation

13.56.2.1 boundingBoxMax

```
ZappyTypes::Vector3 Model3D::boundingBoxMax
```

Definition at line 29 of file ModelManager.hpp.

13.56.2.2 boundingBoxMin

```
ZappyTypes::Vector3 Model3D::boundingBoxMin
```

Definition at line 28 of file ModelManager.hpp.

13.56.2.3 modelId

```
int Model3D::modelId
```

Definition at line 25 of file ModelManager.hpp.

13.56.2.4 scale

`ZappyTypes::Vector3 Model3D::scale`
 Definition at line 27 of file ModelManager.hpp.

13.56.2.5 textureIds

`std::vector<int> Model3D::textureIds`
 Definition at line 26 of file ModelManager.hpp.
 The documentation for this struct was generated from the following file:

- /root/Desktop/Zappy/src/GUI/textureManager/ModelManager.hpp

13.57 ModelManager Class Reference

Gestionnaire de modèles 3D singleton qui implémente un pattern de pool de ressources pour éviter de charger plusieurs fois les mêmes modèles.

```
#include <ModelManager.hpp>
```

Public Member Functions

- void `setGraphicsLib` (`std::shared_ptr< IGraphicsLib > graphicsLib`)
Définit la bibliothèque graphique à utiliser.
- bool `hasModelLibSet` () const
Vérifie si la bibliothèque graphique a été initialisée.
- int `loadModel` (`const std::string &modelPath, const std::string &texturePath=""`)
Charge un modèle 3D à partir d'un chemin Si le modèle existe déjà dans le cache, il est retourné directement.
- int `loadModelWithTextures` (`const std::string &modelPath, const std::vector< std::string > &texturePaths`)
Charge un modèle avec plusieurs textures.
- void `drawModel` (`int modelId, ZappyTypes::Vector3 position, ZappyTypes::Color color=ZappyTypes::Colors::Z_WHITE`)
Dessine un modèle 3D à une position donnée.
- void `drawModelEx` (`int modelId, ZappyTypes::Vector3 position, ZappyTypes::Vector3 rotationAxis, float rotationAngle, float scale=1.0f`)
Dessine un modèle 3D avec rotation.
- void `unloadModel` (`int modelId`)
Libère un modèle 3D spécifique et ses textures associées.
- void `unloadAllModels` ()
Libère tous les modèles 3D en mémoire.
- bool `hasModel` (`const std::string &modelPath`) const
Vérifie si un modèle est déjà chargé
- int `getModelId` (`const std::string &modelPath`) const
Obtient l'ID d'un modèle déjà chargé
- size_t `getModelCount` () const
Obtient le nombre de modèles actuellement chargés.
- std::pair< `ZappyTypes::Vector3, ZappyTypes::Vector3 >` `getBoundingBox` (`int modelId`) const
Obtient la boîte englobante d'un modèle.

Static Public Member Functions

- static `ModelManager & getInstance` ()
Obtient l'instance unique du gestionnaire de modèles.

13.57.1 Detailed Description

Gestionnaire de modèles 3D singleton qui implémente un pattern de pool de ressources pour éviter de charger plusieurs fois les mêmes modèles.

Definition at line 36 of file ModelManager.hpp.

13.57.2 Member Function Documentation

13.57.2.1 drawModel()

```
void ModelManager::drawModel (
    int modelId,
    ZappyTypes::Vector3 position,
    ZappyTypes::Color color = ZappyTypes::Colors::Z_WHITE )
```

Dessine un modèle 3D à une position donnée.

Parameters

<i>modelId</i>	ID du modèle à dessiner
<i>position</i>	Position du modèle dans l'espace 3D
<i>color</i>	Couleur de teinte (blanc par défaut)

Definition at line 82 of file ModelManager.cpp.

```
82
83     std::lock_guard<std::mutex> lock(m_mutex);
84     if (!validateModelForDrawing(modelId))
85         return;
86     m_graphicsLib->DrawModel3D(modelId, position, 1.0f, color);
87 }
```

{

13.57.2.2 drawModelEx()

```
void ModelManager::drawModelEx (
    int modelId,
    ZappyTypes::Vector3 position,
    ZappyTypes::Vector3 rotationAxis,
    float rotationAngle,
    float scale = 1.0f )
```

Dessine un modèle 3D avec rotation.

Parameters

<i>modelId</i>	ID du modèle à dessiner
<i>position</i>	Position du modèle
<i>rotationAxis</i>	Axe de rotation
<i>rotationAngle</i>	Angle de rotation en degrés
<i>scale</i>	Échelle du modèle (1.0f par défaut)

Definition at line 89 of file ModelManager.cpp.

```
91     {
92     std::lock_guard<std::mutex> lock(m_mutex);
93
94     if (!validateModelForDrawing(modelId))
95         return;
96     m_graphicsLib->DrawModelEx(modelId, position, rotationAxis, rotationAngle, scale);
97 }
```

13.57.2.3 getBoundingBox()

```
std::pair< ZappyTypes::Vector3, ZappyTypes::Vector3 > ModelManager::getBoundingBox ( int modelId ) const
```

Obtient la boîte englobante d'un modèle.

Parameters

<i>modelId</i>	ID du modèle
----------------	--------------

Returns

Paire de vecteurs min et max de la boîte englobante

Definition at line 143 of file ModelManager.cpp.

```
143
144     std::lock_guard<std::mutex> lock(m_mutex);
145
146     auto it = m_models.find(modelId);
147     if (it != m_models.end()) {
148         return {it->second.boundingBoxMin, it->second.boundingBoxMax};
149     }
150     return {ZappyTypes::Vector3{0, 0, 0}, ZappyTypes::Vector3{0, 0, 0}};
151 }
```

13.57.2.4 getInstance()

```
static ModelManager& ModelManager::getInstance ( ) [inline], [static]
```

Obtient l'instance unique du gestionnaire de modèles.

Returns

L'instance du [ModelManager](#)

Definition at line 42 of file ModelManager.hpp.

```
42
43     static ModelManager instance;
44     return instance;
45 }
```

13.57.2.5 getModelCount()

```
size_t ModelManager::getModelCount ( ) const
```

Obtient le nombre de modèles actuellement chargés.

Returns

Nombre de modèles en cache

Definition at line 138 of file ModelManager.cpp.

```
138
139     std::lock_guard<std::mutex> lock(m_mutex);
140     return m_models.size();
141 }
```

13.57.2.6 getModelId()

```
int ModelManager::getModelId (
    const std::string & modelPath ) const
```

Obtient l'ID d'un modèle déjà chargé

Parameters

<i>modelPath</i>	Chemin d'accès au fichier du modèle
------------------	-------------------------------------

Returns

ID du modèle ou -1 si non trouvé

Definition at line 129 of file ModelManager.cpp.

```
129
130     std::lock_guard<std::mutex> lock(m_mutex);
131     auto it = m_pathToId.find(modelPath);
132     if (it != m_pathToId.end()) {
133         return it->second;
134     }
135     return -1;
136 }
```

13.57.2.7 hasModel()

```
bool ModelManager::hasModel (
    const std::string & modelPath ) const
```

Vérifie si un modèle est déjà chargé

Parameters

<i>modelPath</i>	Chemin d'accès au fichier du modèle
------------------	-------------------------------------

Returns

true si le modèle existe, false sinon

Definition at line 124 of file ModelManager.cpp.

```
124
125     std::lock_guard<std::mutex> lock(m_mutex);
126     return m_pathToId.find(modelPath) != m_pathToId.end();
127 }
```

13.57.2.8 hasModelLibSet()

```
bool ModelManager::hasModelLibSet ( ) const [inline]
```

Vérifie si la bibliothèque graphique a été initialisée.

Returns

true si initialisée, false sinon

Definition at line 57 of file ModelManager.hpp.

```
57 { return m_graphicsLib != nullptr; }
```

13.57.2.9 loadModel()

```
int ModelManager::loadModel (
    const std::string & modelPath,
    const std::string & texturePath = "" )
```

Charge un modèle 3D à partir d'un chemin Si le modèle existe déjà dans le cache, il est retourné directement.

Parameters

<i>modelPath</i>	Chemin d'accès au fichier du modèle
<i>texturePath</i>	Chemin d'accès à la texture principale (optionnel)

Returns

ID du modèle chargé ou -1 si échec

Definition at line 20 of file ModelManager.cpp.

```

20
21     std::lock_guard<std::mutex> lock(m_mutex);
22     auto it = m_pathToId.find(modelPath);
23     if (it != m_pathToId.end()) {
24         return it->second;
25     }
26     if (!m_graphicsLib) {
27         std::cerr << "Erreur: GraphicsLib non initialisé dans ModelManager" << std::endl;
28         return -1;
29     }
30
31     try {
32         int modelId;
33         if (!texturePath.empty())
34             modelId = m_graphicsLib->LoadModelWithTexture(modelPath, texturePath);
35         else
36             modelId = m_graphicsLib->LoadModel3D(modelPath);
37         if (modelId < 0) {
38             std::cerr << "Erreur: Échec du chargement du modèle " << modelPath << std::endl;
39             return -1;
40         }
41         Model3D model = createModelObject(modelId);
42         if (!texturePath.empty()) {
43             auto [success, updatedModel] = loadTextureForModel(model, texturePath);
44             if (success)
45                 model = updatedModel;
46         }
47         registerModel(modelPath, modelId, model);
48         return modelId;
49     } catch (const std::exception& e) {
50         std::cerr << "Erreur lors du chargement du modèle " << modelPath << ":" << e.what() << std::endl;
51         return -1;
52     }
53 }
```

13.57.2.10 loadModelWithTextures()

```
int ModelManager::loadModelWithTextures (
    const std::string & modelPath,
    const std::vector< std::string > & texturePaths )
```

Charge un modèle avec plusieurs textures.

Parameters

<i>modelPath</i>	Chemin d'accès au fichier du modèle
<i>texturePaths</i>	Liste des chemins d'accès aux textures

Returns

ID du modèle chargé ou -1 si échec

Definition at line 55 of file ModelManager.cpp.

```

55
56     {
57     std::lock_guard<std::mutex> lock(m_mutex);
58     int cachedModelId = checkModelCache(modelPath);
59     if (cachedModelId != -1)
60         return cachedModelId;
61     if (!checkGraphicsLibInitialized())
62         return -1;
63     try {
64         int modelId;
65         if (!texturePaths.empty())
66             modelId = m_graphicsLib->LoadModelWithTexture(modelPath, texturePaths[0]);
67         else
68             modelId = m_graphicsLib->LoadModel3D(modelPath);
69         if (modelId < 0) {
70             std::cerr << "Erreur: Échec du chargement du modèle " << modelPath << std::endl;
71             return -1;
72         }
73         Model3D model = createModelObject(modelId);
74         model = loadTexturesForModel(model, texturePaths);
```

```

74     registerModel(modelPath, modelId, model);
75     return modelId;
76 } catch (const std::exception& e) {
77     std::cerr << "Erreur lors du chargement du modèle " << modelPath << ":" << e.what() << std::endl;
78     return -1;
79 }
80 }
```

13.57.2.11 setGraphicsLib()

```
void ModelManager::setGraphicsLib (
    std::shared_ptr< IGraphicsLib > graphicsLib )
```

Définit la bibliothèque graphique à utiliser.

Parameters

<i>graphicsLib</i>	Pointeur partagé vers une implémentation de IGraphicsLib
--------------------	--

Definition at line 15 of file ModelManager.cpp.

```

15
16     std::lock_guard<std::mutex> lock(m_mutex);
17     m_graphicsLib = graphicsLib;
18 }
```

13.57.2.12 unloadAllModels()

```
void ModelManager::unloadAllModels ( )
```

Libère tous les modèles 3D en mémoire.

Definition at line 113 of file ModelManager.cpp.

```

113     {
114     std::lock_guard<std::mutex> lock(m_mutex);
115
116     for (const auto& [modelId, model] : m_models) {
117         unloadModelTextures(model);
118         unloadModelFromGraphicsLib(modelId);
119     }
120     m_models.clear();
121     m_pathToId.clear();
122 }
```

13.57.2.13 unloadModel()

```
void ModelManager::unloadModel (
    int modelId )
```

Libère un modèle 3D spécifique et ses textures associées.

Parameters

<i>modelId</i>	ID du modèle à libérer
----------------	------------------------

Definition at line 99 of file ModelManager.cpp.

```

99
100    std::lock_guard<std::mutex> lock(m_mutex);
101    auto modelIt = m_models.find(modelId);
102
103    if (modelIt == m_models.end()) {
104        std::cerr << "Tentative de libération d'un modèle inexistant (ID: " << modelId << ")" << std::endl;
105        return;
106    }
107    unloadModelTextures(modelIt->second);
108    unloadModelFromGraphicsLib(modelId);
109    removeModelPathReferences(modelId);
110    m_models.erase(modelIt);
111 }
```

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

- /root/Desktop/Zappy/src/GUI/textureManager/ModelManager.hpp
- /root/Desktop/Zappy/src/GUI/textureManager/ModelManager.cpp

13.58 Zappy::ModelManagerAdapter Class Reference

Adaptateur qui permet de gérer [ModelManager](#) comme un std::shared_ptr. Cette classe encapsule le singleton [ModelManager](#) pour l'utiliser avec des shared_ptr.

```
#include <ModelManagerAdapter.hpp>
```

Public Member Functions

- [ModelManagerAdapter \(\)](#)
Constructeur qui encapsule une référence au singleton [ModelManager](#).
- [ModelManager & getManager \(\)](#)
Obtient une référence au singleton [ModelManager](#) encapsulé
- void [setGraphicsLib \(std::shared_ptr< IGraphicsLib > graphicsLib\)](#)
Définit la bibliothèque graphique à utiliser.

Static Public Member Functions

- static std::shared_ptr< [ModelManagerAdapter](#) > [createShared \(\)](#)
Fonction utilitaire pour créer un shared_ptr à [ModelManager](#).

13.58.1 Detailed Description

Adaptateur qui permet de gérer [ModelManager](#) comme un std::shared_ptr. Cette classe encapsule le singleton [ModelManager](#) pour l'utiliser avec des shared_ptr.

Definition at line 19 of file ModelManagerAdapter.hpp.

13.58.2 Constructor & Destructor Documentation

13.58.2.1 ModelManagerAdapter()

```
Zappy::ModelManagerAdapter::ModelManagerAdapter ( ) [inline], [explicit]
Constructeur qui encapsule une référence au singleton ModelManager.
Definition at line 27 of file ModelManagerAdapter.hpp.
27 : managerRef(ModelManager::getInstance()) {}
```

13.58.3 Member Function Documentation

13.58.3.1 createShared()

```
static std::shared_ptr<ModelManagerAdapter> Zappy::ModelManagerAdapter::createShared ( )
[inline], [static]
Fonction utilitaire pour créer un shared_ptr à ModelManager.
```

Returns

shared_ptr vers [ModelManagerAdapter](#)

Definition at line 39 of file ModelManagerAdapter.hpp.

```
39
40     return std::make_shared<ModelManagerAdapter>();
41 }
```

13.58.3.2 getManager()

```
ModelManager& Zappy::ModelManagerAdapter::getManager () [inline]
Obtient une référence au singleton ModelManager encapsulé
```

Returns

Référence à [ModelManager](#)

Definition at line 33 of file ModelManagerAdapter.hpp.

```
33 { return managerRef; }
```

13.58.3.3 setGraphicsLib()

```
void Zappy::ModelManagerAdapter::setGraphicsLib (
    std::shared_ptr< IGraphicsLib > graphicsLib ) [inline]
```

Définit la bibliothèque graphique à utiliser.

Parameters

<i>graphicsLib</i>	Pointeur partagé vers une implémentation de IGraphicsLib
--------------------	--

Definition at line 49 of file ModelManagerAdapter.hpp.

```
49
50     managerRef.setGraphicsLib(graphicsLib);
51 }
```

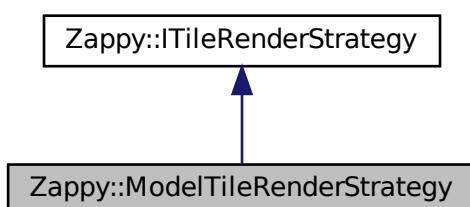
The documentation for this class was generated from the following file:

- /root/Desktop/Zappy/src/GUI/textureManager/[ModelManagerAdapter.hpp](#)

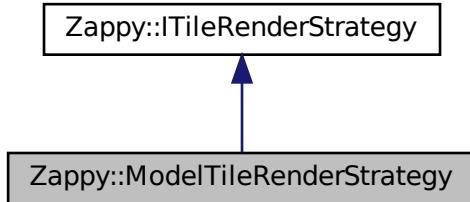
13.59 Zappy::ModelTileRenderStrategy Class Reference

```
#include <ModelTileRenderStrategy.hpp>
```

Inheritance diagram for Zappy::ModelTileRenderStrategy:



Collaboration diagram for Zappy::ModelTileRenderStrategy:



Public Member Functions

- `ModelTileRenderStrategy` (const std::shared_ptr< `ModelManager` > &manager, int modelId, const std::shared_ptr< `GraphicalContext` > &ctx)
- void `renderTile` (const std::shared_ptr< `IGraphicsLib` > &graphicsLib, int x, int y, const `ZappyTypes::Color` &color, float tileSize, float spacing) override

13.59.1 Detailed Description

Definition at line 16 of file ModelTileRenderStrategy.hpp.

13.59.2 Constructor & Destructor Documentation

13.59.2.1 ModelTileRenderStrategy()

```
Zappy::ModelTileRenderStrategy::ModelTileRenderStrategy (
    const std::shared_ptr< ModelManager > & manager,
    int modelId,
    const std::shared_ptr< GraphicalContext > & ctx )
Definition at line 16 of file ModelTileRenderStrategy.cpp.
17      : modelManager(manager), modelId(id), context(ctx) {}
```

13.59.3 Member Function Documentation

13.59.3.1 renderTile()

```
void Zappy::ModelTileRenderStrategy::renderTile (
    const std::shared_ptr< IGraphicsLib > & graphicsLib,
    int x,
    int y,
    const ZappyTypes::Color & color,
    float tileSize,
    float spacing ) [override], [virtual]
```

Implements `Zappy::ITileRenderStrategy`.

Definition at line 19 of file ModelTileRenderStrategy.cpp.

```
23      {
24      float mapOffset = context->getMapWidth() / 2.0f;
25      ZappyTypes::Vector3 position = {
26          (x - mapOffset + 0.5f) * (tileSize + spacing),
```

```

27     0.0f,
28     (y - mapOffset + 0.5f) * (tileSize + spacing)
29   };
30   modelManager->drawModel(modelId, position, color);
31   ZappyTypes::Color baseColor = {50, 50, 50, 200};
32   graphicsLib->DrawCube({position.x, position.y - 0.05f, position.z},
33                         tileSize, 0.01f, tileSize, baseColor);
34   ZappyTypes::Color borderColor = {100, 100, 100, 255};
35   float halfSize = tileSize / 2;
36   graphicsLib->DrawLine3D({position.x - halfSize, position.y - 0.05f, position.z - halfSize},
37                           {position.x + halfSize, position.y - 0.05f, position.z - halfSize},
38                           borderColor);
39   graphicsLib->DrawLine3D({position.x + halfSize, position.y - 0.05f, position.z - halfSize},
40                           {position.x + halfSize, position.y - 0.05f, position.z + halfSize},
41                           borderColor);
42   graphicsLib->DrawLine3D({position.x + halfSize, position.y - 0.05f, position.z + halfSize},
43                           {position.x - halfSize, position.y - 0.05f, position.z + halfSize},
44                           borderColor);
45   graphicsLib->DrawLine3D({position.x - halfSize, position.y - 0.05f, position.z + halfSize},
46                           {position.x - halfSize, position.y - 0.05f, position.z - halfSize},
47                           borderColor);
48 }

```

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

- /root/Desktop/Zappy/src/GUI/renderer/strategies/ModelTileRenderStrategy.hpp
- /root/Desktop/Zappy/src/GUI/renderer/strategies/ModelTileRenderStrategy.cpp

13.60 raylibcpp::ModelWrap Class Reference

```
#include <Texture.hpp>
```

Public Member Functions

- [ModelWrap](#) (const std::string &objPath)
- [~ModelWrap](#) ()
- void [draw](#) (Vector3 position, float scale=1.0f, Color tint=WHITE) const
- Model & [get](#) ()

13.60.1 Detailed Description

Definition at line 39 of file Texture.hpp.

13.60.2 Constructor & Destructor Documentation

13.60.2.1 ModelWrap()

```
raylibcpp::ModelWrap::ModelWrap (
    const std::string & objPath ) [explicit]
```

Definition at line 63 of file Texture.cpp.

```

63
64   model = LoadModel(objPath.c_str());
65   if (model.meshCount == 0)
66     throw std::runtime_error("Erreur chargement model: " + objPath);
67   bool hasTextures = false;
68   for (int i = 0; i < model.materialCount; i++) {
69     if (model.materials[i].maps[MATERIAL_MAP_DIFFUSE].texture.id > 0) {
70       hasTextures = true;
71       break;
72     }
73   }
74
75   if (!hasTextures && model.materialCount > 0) {
76     std::cout << "Aucune texture chargée automatiquement pour: " << objPath << std::endl;
77   }
78 }
```

13.60.2.2 ~ModelWrap()

```
raylibcpp::ModelWrap::~ModelWrap ( )
Definition at line 80 of file Texture.cpp.
80
81     UnloadModel(model);
82 }
```

13.60.3 Member Function Documentation

13.60.3.1 draw()

```
void raylibcpp::ModelWrap::draw (
    Vector3 position,
    float scale = 1.0f,
    Color tint = WHITE ) const
Definition at line 84 of file Texture.cpp.
84
85     DrawModel(model, position, scale, tint);
86 }
```

13.60.3.2 get()

```
Model & raylibcpp::ModelWrap::get ( )
Definition at line 88 of file Texture.cpp.
88
89     return model;
90 }
```

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

- /root/Desktop/Zappy/libs/RaylibCPP/texture/[Texture.hpp](#)
- /root/Desktop/Zappy/libs/RaylibCPP/texture/[Texture.cpp](#)

13.61 raylibcpp::MusicWrap Class Reference

```
#include <Audio.hpp>
```

Public Member Functions

- [MusicWrap](#) (const std::string &path)
- [~MusicWrap](#) ()
- void [play](#) () const
- void [stop](#) () const
- void [pause](#) () const
- void [resume](#) () const
- void [update](#) () const
- void [setVolume](#) (float volume) const
- Music & [get](#) ()

13.61.1 Detailed Description

Definition at line 34 of file Audio.hpp.

13.61.2 Constructor & Destructor Documentation

13.61.2.1 MusicWrap()

```
raylibcpp::MusicWrap::MusicWrap (
    const std::string & path) [explicit]
Definition at line 56 of file Audio.cpp.
56
57     music = LoadMusicStream(path.c_str());
58     if (music.ctxData == nullptr)
59         throw std::runtime_error("Erreur chargement music: " + path);
60 }
```

13.61.2.2 ~MusicWrap()

```
raylibcpp::MusicWrap::~MusicWrap()
Definition at line 62 of file Audio.cpp.
62
63     UnloadMusicStream(music);
64 }
```

13.61.3 Member Function Documentation

13.61.3.1 get()

```
Music & raylibcpp::MusicWrap::get()
Definition at line 90 of file Audio.cpp.
90
91     return music;
92 }
```

13.61.3.2 pause()

```
void raylibcpp::MusicWrap::pause() const
Definition at line 74 of file Audio.cpp.
74
75     PauseMusicStream(music);
76 }
```

13.61.3.3 play()

```
void raylibcpp::MusicWrap::play() const
Definition at line 66 of file Audio.cpp.
66
67     PlayMusicStream(music);
68 }
```

13.61.3.4 resume()

```
void raylibcpp::MusicWrap::resume() const
Definition at line 78 of file Audio.cpp.
78
79     ResumeMusicStream(music);
80 }
```

13.61.3.5 setVolume()

```
void raylibcpp::MusicWrap::setVolume(
    float volume) const
Definition at line 86 of file Audio.cpp.
86
87     SetMusicVolume(music, volume);
```

```
88 }
```

13.61.3.6 stop()

```
void raylibcpp::MusicWrap::stop ( ) const
Definition at line 70 of file Audio.cpp.
70
71     StopMusicStream(music);
72 }
```

13.61.3.7 update()

```
void raylibcpp::MusicWrap::update ( ) const
Definition at line 82 of file Audio.cpp.
82
83     UpdateMusicStream(music);
84 }
```

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

- /root/Desktop/Zappy/libs/RaylibCPP/audio/[Audio.hpp](#)
- /root/Desktop/Zappy/libs/RaylibCPP/audio/[Audio.cpp](#)

13.62 NetworkLogger Class Reference

```
#include <NetworkLogger.hpp>
```

Public Member Functions

- void [setEnabled](#) (bool enabled)
- void [log](#) (const std::string &msg)
- bool [isEnabled](#) () const

Static Public Member Functions

- static [NetworkLogger](#) & [get](#) ()

13.62.1 Detailed Description

Definition at line 6 of file NetworkLogger.hpp.

13.62.2 Member Function Documentation

13.62.2.1 get()

```
static NetworkLogger& NetworkLogger::get ( ) [inline], [static]
Definition at line 8 of file NetworkLogger.hpp.
8
9     static NetworkLogger instance;
10    return instance;
11 }
```

13.62.2.2 isEnabled()

```
bool NetworkLogger::isEnabled ( ) const [inline]
Definition at line 21 of file NetworkLogger.hpp.
21 { return _enabled; }
```

13.62.2.3 log()

```
void NetworkLogger::log (
    const std::string & msg ) [inline]
Definition at line 15 of file NetworkLogger.hpp.
15
16     if (_enabled) {
17         std::lock_guard<std::mutex> lock(_mutex);
18         std::cout << "[DEBUG] " << msg << std::endl;
19     }
20 }
```

13.62.2.4 setEnabled()

```
void NetworkLogger::setEnabled (
    bool enabled ) [inline]
Definition at line 12 of file NetworkLogger.hpp.
12
13     _enabled = enabled;
14 }
```

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

- /root/Desktop/Zappy/src/GUI/network/networkManager/[NetworkLogger.hpp](#)

13.63 NetworkManager Class Reference

```
#include <NetworkManager.hpp>
```

Public Types

- using [MessageCallback](#) = std::function< void(const std::string &, const std::string &) >
- using [ConnectionCallback](#) = std::function< void(bool) >

Public Member Functions

- [NetworkManager \(\)](#)
- [~NetworkManager \(\)](#)
- bool [connect](#) (const std::string &host, int port)
- void [disconnect \(\)](#)
- bool [isConnected \(\)](#) const
- void [sendCommand](#) (const std::string &command)
- void [processIncomingMessages \(\)](#)
- void [setMessageCallback](#) ([MessageCallback](#) callback)
- void [setConnectionCallback](#) ([ConnectionCallback](#) callback)
- void [networkThreadLoop \(\)](#)

13.63.1 Detailed Description

Definition at line 25 of file NetworkManager.hpp.

13.63.2 Member Typedef Documentation

13.63.2.1 ConnectionCallback

```
using NetworkManager::ConnectionCallback = std::function<void(bool)>
Definition at line 40 of file NetworkManager.hpp.
```

13.63.2.2 MessageCallback

```
using NetworkManager::MessageCallback = std::function<void(const std::string&, const std::string&)>
Definition at line 37 of file NetworkManager.hpp.
```

13.63.3 Constructor & Destructor Documentation

13.63.3.1 NetworkManager()

```
NetworkManager::NetworkManager ( )
Definition at line 19 of file NetworkManager.cpp.
20     : _connection(std::make_unique<TcpConnection>()),
21     _protocolParser(std::make_unique<ProtocolParser>()),
22     _networkThread(std::make_unique<NetworkThread>()),
23     _incomingQueue(std::make_unique<MessageQueue>()),
24     _outgoingQueue(std::make_unique<MessageQueue>()),
25     _receiveBuffer(""),
26     _graphicalContext(std::make_unique<GraphicalContext>()),
27     _isConnected(false) {
28 }
```

13.63.3.2 ~NetworkManager()

```
NetworkManager::~NetworkManager ( )
Definition at line 30 of file NetworkManager.cpp.
30     {
31     {
32         std::lock_guard<std::mutex> lock(_mutex);
33         _isConnected = false;
34     }
35     if (_networkThread && _networkThread->isRunning()) {
36         _networkThread->stop();
37     }
38     if (_connection) {
39         _connection->close();
40     }
41 }
```

13.63.4 Member Function Documentation

13.63.4.1 connect()

```
bool NetworkManager::connect (
    const std::string & host,
    int port )
```

Definition at line 43 of file NetworkManager.cpp.

```
43     std::lock_guard<std::mutex> lock(_mutex);
44
45
46     if (_isConnected) {
47         NetworkLogger::get().log(std::string("[INFO] Already connected to ") + host + ":" +
48             std::to_string(port));
49         return true;
50     }
51     try {
52         _connection->connect(host, port);
53         _isConnected = true;
54         _networkThread->start([this]() { this->networkThreadLoop(); });
55         if (_connectionCallback)
56             _connectionCallback(true);
57         return true;
58     } catch (const std::exception& e) {
59         NetworkLogger::get().log(std::string("[ERROR] Connection error: ") + e.what());
60         _isConnected = false;
61         return false;
62     }
```

```
63 }
```

13.63.4.2 disconnect()

```
void NetworkManager::disconnect ( )
Definition at line 65 of file NetworkManager.cpp.
65
66     bool wasConnected = false;
67     {
68         std::lock_guard<std::mutex> lock(_mutex);
69         wasConnected = _isConnected;
70         _isConnected = false;
71     }
72     if (wasConnected) {
73         _networkThread->stop();
74         _connection->close();
75         if (_connectionCallback)
76             _connectionCallback(false);
77     }
78 }
```

13.63.4.3 isConnected()

```
bool NetworkManager::isConnected ( ) const
Definition at line 80 of file NetworkManager.cpp.
80
81     return _isConnected && _connection->isConnected();
82 }
```

13.63.4.4 networkThreadLoop()

```
void NetworkManager::networkThreadLoop ( )
Definition at line 122 of file NetworkManager.cpp.
122
123     int errorCount = 0;
124     const int maxErrors = 3;
125     NetworkLogger::get().log("Network thread started");
126     if (!tryReceiveInitialWelcome()) {
127         NetworkLogger::get().log("Failed to receive WELCOME message, exiting network thread");
128         return;
129     }
130     while (_networkThread->isRunning()) {
131         if (!_isConnected) {
132             NetworkLogger::get().log("Network thread: connection lost, exiting...");
133             break;
134         }
135         try {
136             errorCount = receiveAndProcessData(errorCount, maxErrors);
137             processIncomingMessages();
138             errorCount = processPendingOutgoingMessages(errorCount, maxErrors);
139             std::this_thread::sleep_for(std::chrono::milliseconds(10));
140         } catch (const std::exception& e) {
141             errorCount = handleNetworkThreadError(errorCount, maxErrors, e);
142             if (errorCount >= maxErrors) {
143                 break;
144             }
145         }
146     }
147     NetworkLogger::get().log("Network thread exited");
148 }
```

13.63.4.5 processIncomingMessages()

```
void NetworkManager::processIncomingMessages ( )
Definition at line 268 of file NetworkManager.cpp.
268
269     std::string message;
270     while (!message = _incomingQueue->dequeue()).empty() {
271         NetworkLogger::get().log(std::string("Processing incoming message: ") + message);
272         processIncomingMessage(message);
273     }
274 }
```

13.63.4.6 sendCommand()

```
void NetworkManager::sendCommand (
    const std::string & command )
Definition at line 84 of file NetworkManager.cpp.
84
85     if (!validateConnectionForSending()) {
86         return;
87     }
88     std::string finalCommand = formatCommand(command);
89     logOutgoingCommand(finalCommand);
90     queueCommandForSending(finalCommand);
91 }
```

13.63.4.7 setConnectionCallback()

```
void NetworkManager::setConnectionCallback (
    ConnectionCallback callback )
Definition at line 349 of file NetworkManager.cpp.
349
350     _connectionCallback = callback;
351 }
```

13.63.4.8 setMessageCallback()

```
void NetworkManager::setMessageCallback (
    MessageCallback callback )
Definition at line 345 of file NetworkManager.cpp.
345
346     _messageCallback = callback;
347 }
```

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

- /root/Desktop/Zappy/src/GUI/network/networkManager/[NetworkManager.hpp](#)
- /root/Desktop/Zappy/src/GUI/network/networkManager/[NetworkManager.cpp](#)

13.64 NetworkThread Class Reference

```
#include <NetworkThread.hpp>
```

Public Member Functions

- [NetworkThread \(\)](#)
- [~NetworkThread \(\)](#)
- void [start \(const std::function< void\(\)> &threadFunction\)](#)
- void [stop \(\)](#)
- bool [isRunning \(\) const](#)

13.64.1 Detailed Description

Definition at line 16 of file NetworkThread.hpp.

13.64.2 Constructor & Destructor Documentation

13.64.2.1 NetworkThread()

```
NetworkThread::NetworkThread ( )
Definition at line 10 of file NetworkThread.cpp.
10 : _isRunning(false) {
11 }
```

13.64.2.2 ~NetworkThread()

```
NetworkThread::~NetworkThread ( )
Definition at line 13 of file NetworkThread.cpp.
13 {
14     stop();
15 }
```

13.64.3 Member Function Documentation

13.64.3.1 isRunning()

```
bool NetworkThread::isRunning ( ) const
Definition at line 32 of file NetworkThread.cpp.
32 {
33     return _isRunning;
34 }
```

13.64.3.2 start()

```
void NetworkThread::start (
    const std::function< void()> & threadFunction )
Definition at line 17 of file NetworkThread.cpp.
17 {
18     if (! _isRunning) {
19         _isRunning = true;
20         _thread = std::thread(threadFunction);
21     }
22 }
```

13.64.3.3 stop()

```
void NetworkThread::stop ( )
Definition at line 24 of file NetworkThread.cpp.
24 {
25     if (_isRunning) {
26         _isRunning = false;
27         if (_thread.joinable())
28             _thread.join();
29     }
30 }
```

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

- /root/Desktop/Zappy/src/GUI/network/networkThread/[NetworkThread.hpp](#)
- /root/Desktop/Zappy/src/GUI/network/networkThread/[NetworkThread.cpp](#)

13.65 RayGUICPP::Panel Class Reference

```
#include <RayGuiEncap.hpp>
```

Static Public Member Functions

- static void [Draw](#) (const Rectangle &bounds)

13.65.1 Detailed Description

Definition at line 173 of file RayGuiEncap.hpp.

13.65.2 Member Function Documentation

13.65.2.1 Draw()

```
static void RayGUICPP::Panel::Draw (
    const Rectangle & bounds ) [inline], [static]
Definition at line 175 of file RayGuiEncap.hpp.
175
176     GuiPanel(bounds, nullptr);
177 }
```

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

- /root/Desktop/Zappy/libs/RayGUICPP/include/RayGuiEncap.hpp

13.66 paringArgsClass.parseArgs Class Reference

Public Member Functions

- def `__init__` (self)
- def `hasFlagAndValue` (self, flag)
- def `getFlagValue` (self, flag)
- def `searchUnknownFlag` (self, flag)
- def `parseArgs` (self)
- def `getPort` (self)
- def `setPort` (self, port)
- def `getName` (self)
- def `setName` (self, name)
- def `getMachine` (self)
- def `setMachine` (self, machine)

Data Fields

- args
- port
- name
- machine
- parsed_args

13.66.1 Detailed Description

Definition at line 10 of file paringArgsClass.py.

13.66.2 Constructor & Destructor Documentation

13.66.2.1 __init__()

```
def paringArgsClass.parseArgs.__init__ (
    self )
Definition at line 11 of file paringArgsClass.py.
11     def __init__(self):
12         self.args = sys.argv[1:]
13         self.port = 0
14         self.name = ""
15         self.machine = "localhost"
16         try:
17             if (len(self.args) == 0):
18                 raise ParsingArgsError("No arguments provided. Use -p <port> -n <name> -h <machine> to
specify the required parameters.")
19             self.parsed_args = self.parseArgs()
20         except ParsingArgsError:
21             sys.exit(84)
22
23
```

13.66.2.2 parseArgs()

```
def paringArgsClass.parseArgs.parseArgs (
    self )

```

Definition at line 49 of file paringArgsClass.py.

```
49     def parseArgs(self):
50         # add flags if wanted
51         self.searchUnknownFlag(["-p", "-n", "-h"])
52         # maybe to a for loop to check if the flags are in the args
53         if (self.hasFlagAndValue("-p")):
54             self.setPort(self.getFlagValue("-p"))
55         else:
56             raise ParsingArgsError("No port specified. Use -p <port> to specify a port.")
57
58         if (self.hasFlagAndValue("-n")):
59             self.setName(self.getFlagValue("-n"))
60         else:
61             raise ParsingArgsError("No name specified. Use -n <name> to specify a name.")
62
63         if (self.hasFlagAndValue("-h")):
64             self.setMachine(self.getFlagValue("-h"))
65         else:
66             print("No machine specified, using default machine 'localhost'")
67
```

13.66.3 Member Function Documentation**13.66.3.1 getFlagValue()**

```
def paringArgsClass.parseArgs.getFlagValue (
    self,
    flag )

```

Definition at line 36 of file paringArgsClass.py.

```
36     def getFlagValue(self, flag):
37         for i, arg in enumerate(self.args):
38             if arg == flag and i + 1 < len(self.args):
39                 return self.args[i + 1]
40
41         return None
```

13.66.3.2 getMachine()

```
def paringArgsClass.parseArgs.getMachine (
    self )

```

Definition at line 88 of file paringArgsClass.py.

```
88     def getMachine(self):
89         return self.machine
90
```

13.66.3.3 getName()

```
def paringArgsClass.parseArgs.getName (
    self )
Definition at line 80 of file paringArgsClass.py.
80     def getName(self):
81         return self.name
82
```

13.66.3.4 getPort()

```
def paringArgsClass.parseArgs.getPort (
    self )
Definition at line 68 of file paringArgsClass.py.
68     def getPort(self):
69         return self.port
70
```

13.66.3.5 hasFlagAndValue()

```
def paringArgsClass.parseArgs.hasFlagAndValue (
    self,
    flag )
Definition at line 24 of file paringArgsClass.py.
24     def hasFlagAndValue(self, flag):
25         for i, arg in enumerate(self.args):
26             if arg == flag:
27                 if i + 1 < len(self.args):
28                     if not self.args[i + 1].startswith('-'):
29                         return True
30                 else:
31                     raise ParsingArgsError(f"Flag {flag} requires a value, but got another flag:
{self.args[i + 1]}")
32             else:
33                 raise ParsingArgsError(f"Flag {flag} requires a value.")
34     return False
35
```

13.66.3.6 searchUnknownFlag()

```
def paringArgsClass.parseArgs.searchUnknownFlag (
    self,
    flag )
Definition at line 42 of file paringArgsClass.py.
42     def searchUnknownFlag(self, flag):
43         i = 0
44         while (i < len(self.args)):
45             if self.args[i] not in flag:
46                 raise ParsingArgsError(f"Unknown argument: {self.args[i]}")
47             i += 2
48
```

13.66.3.7 setMachine()

```
def paringArgsClass.parseArgs.setMachine (
    self,
    machine )
Definition at line 91 of file paringArgsClass.py.
91     def setMachine(self, machine):
92         if not isinstance(machine, str):
93             raise ParsingArgsError("Machine must be a string.")
94         self.machine = machine
```

13.66.3.8 setName()

```
def paringArgsClass.parseArgs.setName (
    self,
    name )
```

Definition at line 83 of file paringArgsClass.py.

```
83     def setName(self, name):
84         if not isinstance(name, str):
85             raise ParsingArgsError("Name must be a string.")
86         self.name = name
87
```

13.66.3.9 setPort()

```
def paringArgsClass.parseArgs.setPort (
    self,
    port )
```

Definition at line 71 of file paringArgsClass.py.

```
71     def setPort(self, port):
72         try:
73             port_int = int(port)
74             if (port_int < 1 or port_int > 65535):
75                 raise ParsingArgsError("Invalid port number. Port must be between 1 and 65535.")
76             self.port = port_int
77         except ValueError:
78             raise ParsingArgsError(f"Invalid port format: {port}. Port must be a number.")
79
```

13.66.4 Field Documentation**13.66.4.1 args**

paringArgsClass.parseArgs.args

Definition at line 12 of file paringArgsClass.py.

13.66.4.2 machine

paringArgsClass.parseArgs.machine

Definition at line 15 of file paringArgsClass.py.

13.66.4.3 name

paringArgsClass.parseArgs.name

Definition at line 14 of file paringArgsClass.py.

13.66.4.4 parsed_args

paringArgsClass.parseArgs.parsed_args

Definition at line 19 of file paringArgsClass.py.

13.66.4.5 port

paringArgsClass.parseArgs.port

Definition at line 13 of file paringArgsClass.py.

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

- /root/Desktop/Zappy/src/Al/parser/[paringArgsClass.py](#)

13.67 parsing_info_s Struct Reference

```
#include <parsing.h>
```

Data Fields

- int [port](#)
- int [width](#)
- int [height](#)
- char ** [names](#)
- int [client_nb](#)
- int [frequence](#)

13.67.1 Detailed Description

Definition at line 12 of file `parsing.h`.

13.67.2 Field Documentation

13.67.2.1 client_nb

```
int parsing_info_s::client_nb
```

Definition at line 17 of file `parsing.h`.

13.67.2.2 frequence

```
int parsing_info_s::frequence
```

Definition at line 18 of file `parsing.h`.

13.67.2.3 height

```
int parsing_info_s::height
```

Definition at line 15 of file `parsing.h`.

13.67.2.4 names

```
char** parsing_info_s::names
```

Definition at line 16 of file `parsing.h`.

13.67.2.5 port

```
int parsing_info_s::port
```

Definition at line 13 of file `parsing.h`.

13.67.2.6 width

```
int parsing_info_s::width
```

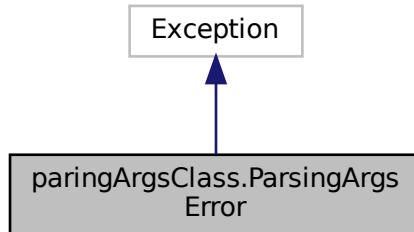
Definition at line 14 of file `parsing.h`.

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

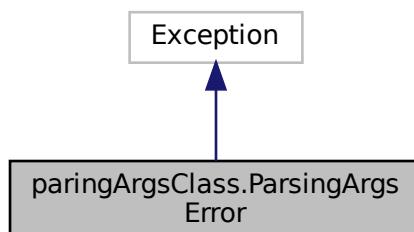
- `/root/Desktop/Zappy/src/Server/include/parsing.h`

13.68 paringArgsClass.ParsingArgsError Class Reference

Inheritance diagram for paringArgsClass.ParsingArgsError:



Collaboration diagram for paringArgsClass.ParsingArgsError:



Public Member Functions

- def [__init__](#) (self, message="Erreur lors du parsing des arguments")

Data Fields

- [message](#)

13.68.1 Detailed Description

Definition at line 4 of file paringArgsClass.py.

13.68.2 Constructor & Destructor Documentation

13.68.2.1 [__init__\(\)](#)

```
def paringArgsClass.ParsingArgsError.__init__ (
    self,
    message = "Erreur lors du parsing des arguments" )
```

Definition at line 5 of file paringArgsClass.py.

```
5     def __init__(self, message="Erreur lors du parsing des arguments"):
6         self.message = message
7         print(f"ParsingArgsError: {message}", file=sys.stderr)
8         super().__init__(self.message)
9 
```

13.68.3 Field Documentation

13.68.3.1 message

paringArgsClass.ParsingArgsError.message

Definition at line 6 of file paringArgsClass.py.

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

- /root/Desktop/Zappy/src/Al/parser/[paringArgsClass.py](#)

13.69 ParsingCLI Class Reference

```
#include <ParsingCLI.hpp>
```

Public Member Functions

- [ParsingCLI](#) (int argc, char **argv)
- [~ParsingCLI](#) ()
- void [parse](#) (int argc, char **argv)
- int [getPort](#) () const
- const std::string & [getMachine](#) () const
- void [setPort](#) (int port)
- void [setMachine](#) (const std::string &machine)
- void [displayHelp](#) () const
- void [checkValidityCLI](#) () const

13.69.1 Detailed Description

Definition at line 14 of file ParsingCLI.hpp.

13.69.2 Constructor & Destructor Documentation

13.69.2.1 ParsingCLI()

```
ParsingCLI::ParsingCLI (
    int argc,
    char ** argv )
```

Definition at line 13 of file ParsingCLI.cpp.

```
13                                         {
14     _port = 0;
15     parse(argc, argv);
16 }
```

13.69.2.2 ~ParsingCLI()

```
ParsingCLI::~ParsingCLI ( )
```

Definition at line 18 of file ParsingCLI.cpp.

```
18                                         {
19 }
```

13.69.3 Member Function Documentation

13.69.3.1 checkValidityCLI()

```
void ParsingCLI::checkValidityCLI () const
Definition at line 58 of file ParsingCLI.cpp.
58
59     if (_machine.empty())
60         throw ParsingCLIEException("Machine name cannot be empty");
61     if (_port <= 0)
62         throw ParsingCLIEException("Port number must be set");
63 }
```

13.69.3.2 displayHelp()

```
void ParsingCLI::displayHelp () const
Definition at line 65 of file ParsingCLI.cpp.
65
66     std::cout << "USAGE: ./zappy_gui -p port -h machine" << std::endl;
67     std::cout << "\t-p port      port number" << std::endl;
68     std::cout << "\t-h machine   hostname of the server" << std::endl;
69 }
```

13.69.3.3 getMachine()

```
const std::string & ParsingCLI::getMachine () const
Definition at line 24 of file ParsingCLI.cpp.
24
25     return _machine;
26 }
```

13.69.3.4 getPort()

```
int ParsingCLI::getPort () const
Definition at line 21 of file ParsingCLI.cpp.
21
22     return _port;
23 }
```

13.69.3.5 parse()

```
void ParsingCLI::parse (
    int argc,
    char ** argv )
Definition at line 40 of file ParsingCLI.cpp.
40
41     if (argc == 1) {
42         displayHelp();
43         throw ParsingCLIEException("No arguments provided");
44     }
45     for (int i = 1; i < argc; ++i) {
46         if (strcmp(argv[i], "-p") == 0 && i + 1 < argc) {
47             setPort(std::stoi(argv[i+1]));
48         } else if (strcmp(argv[i], "-h") == 0 && i + 1 < argc) {
49             setMachine(argv[i+1]);
50         } else {
51             displayHelp();
52             throw ParsingCLIEException("Invalid argument: " + std::string(argv[i]));
53         }
54     }
55     checkValidityCLI();
56 }
```

13.69.3.6 setMachine()

```
void ParsingCLI::setMachine (
    const std::string & machine )
Definition at line 34 of file ParsingCLI.cpp.
34
35     if (!machine.empty() && machine[0] == '-')
36         throw ParsingCLIEException("Invalid machine name starting with '-'");
37     _machine = machine;
38 }
```

13.69.3.7 setPort()

```
void ParsingCLI::setPort (
    int port )
Definition at line 28 of file ParsingCLI.cpp.
28
29     if (port < 1 || port > 65535)
30         throw ParsingCLIEException("Port number must be between 1 and 65535");
31     _port = port;
32 }
```

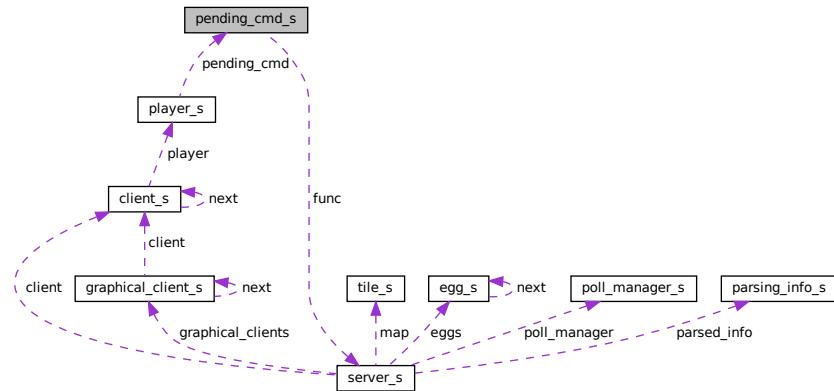
The documentation for this class was generated from the following files:

- /root/Desktop/Zappy/src/GUI/parsing/ParsingCLI.hpp
- /root/Desktop/Zappy/src/GUI/parsing/ParsingCLI.cpp

13.70 pending_cmd_s Struct Reference

```
#include <pending_cmd.h>
```

Collaboration diagram for pending_cmd_s:



Data Fields

- `cmd_func_t func`
- `char * args`

13.70.1 Detailed Description

Definition at line 17 of file pending_cmd.h.

13.70.2 Field Documentation

13.70.2.1 args

```
char* pending_cmd_s::args  
Definition at line 19 of file pending_cmd.h.
```

13.70.2.2 func

```
cmd_func_t pending_cmd_s::func  
Definition at line 18 of file pending_cmd.h.  
The documentation for this struct was generated from the following file:
```

- /root/Desktop/Zappy/src/Server/include/pending_cmd.h

13.71 player_inventory_s Struct Reference

```
#include <player.h>
```

Data Fields

- [resource_type_t type](#)
- [char * name](#)
- [int quantity](#)

13.71.1 Detailed Description

Definition at line 21 of file player.h.

13.71.2 Field Documentation

13.71.2.1 name

```
char* player_inventory_s::name  
Definition at line 23 of file player.h.
```

13.71.2.2 quantity

```
int player_inventory_s::quantity  
Definition at line 24 of file player.h.
```

13.71.2.3 type

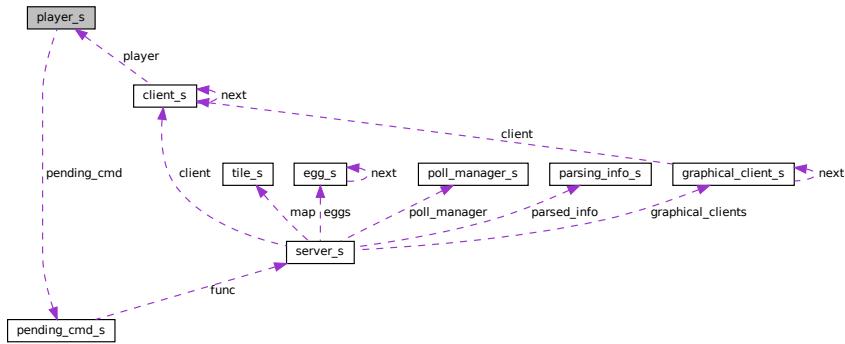
```
resource_type_t player_inventory_s::type  
Definition at line 22 of file player.h.  
The documentation for this struct was generated from the following file:
```

- /root/Desktop/Zappy/src/Server/include/player.h

13.72 player_s Struct Reference

```
#include <player.h>
```

Collaboration diagram for player_s:



Data Fields

- int `pos_x`
- int `pos_y`
- enum `rotation_e` `rotation`
- int `level`
- int `life`
- char * `team_name`
- int `inventory` [COUNT]
- int `inventory_size`
- int `busy_until`
- char ** `command_queue`
- int `queue_size`
- bool `is_in_incantation`
- int `incantation_leader_id`
- `pending_cmd_t` * `pending_cmd`

13.72.1 Detailed Description

Definition at line 27 of file `player.h`.

13.72.2 Field Documentation

13.72.2.1 `busy_until`

`int player_s::busy_until`

Definition at line 36 of file `player.h`.

13.72.2.2 `command_queue`

`char** player_s::command_queue`

Definition at line 37 of file `player.h`.

13.72.2.3 incantation_leader_id

```
int player_s::incantation_leader_id
```

Definition at line 40 of file player.h.

13.72.2.4 inventory

```
int player_s::inventory[COUNT]
```

Definition at line 34 of file player.h.

13.72.2.5 inventory_size

```
int player_s::inventory_size
```

Definition at line 35 of file player.h.

13.72.2.6 is_in_incantation

```
bool player_s::is_in_incantation
```

Definition at line 39 of file player.h.

13.72.2.7 level

```
int player_s::level
```

Definition at line 31 of file player.h.

13.72.2.8 life

```
int player_s::life
```

Definition at line 32 of file player.h.

13.72.2.9 pending_cmd

```
pending_cmd_t* player_s::pending_cmd
```

Definition at line 41 of file player.h.

13.72.2.10 pos_x

```
int player_s::pos_x
```

Definition at line 28 of file player.h.

13.72.2.11 pos_y

```
int player_s::pos_y
```

Definition at line 29 of file player.h.

13.72.2.12 queue_size

```
int player_s::queue_size
```

Definition at line 38 of file player.h.

13.72.2.13 rotation

```
enum rotation_e player_s::rotation
Definition at line 29 of file player.h.
```

13.72.2.14 team_name

```
char* player_s::team_name
Definition at line 33 of file player.h.
```

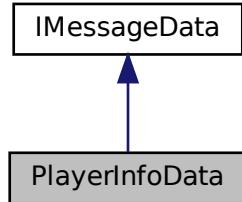
The documentation for this struct was generated from the following file:

- /root/Desktop/Zappy/src/Server/include/player.h

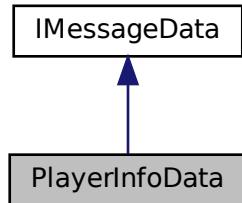
13.73 PlayerInfoData Class Reference

```
#include <PlayerInfoData.hpp>
```

Inheritance diagram for PlayerInfoData:



Collaboration diagram for PlayerInfoData:



Public Member Functions

- `PlayerInfoData ()`
- `PlayerInfoData (int id, int x, int y, int orientation, int level, const std::string &teamName="")`
- `MessageType getType () const override`
- `int getId () const`
- `int getX () const`

- int `getY () const`
- int `getOrientation () const`
- int `getLevel () const`
- const std::string & `getTeamName () const`
- void `setId (int value)`
- void `setX (int value)`
- void `setY (int value)`
- void `setOrientation (int value)`
- void `setLevel (int value)`
- void `setTeamName (const std::string &value)`

13.73.1 Detailed Description

Definition at line 14 of file PlayerInfoData.hpp.

13.73.2 Constructor & Destructor Documentation

13.73.2.1 PlayerInfoData() [1/2]

```
PlayerInfoData::PlayerInfoData ( ) [inline]
```

Definition at line 16 of file PlayerInfoData.hpp.

```
16 : _id(0), _x(0), _y(0), _orientation(0), _level(0), _teamName("") {}
```

13.73.2.2 PlayerInfoData() [2/2]

```
PlayerInfoData::PlayerInfoData (   
    int id,  
    int x,  
    int y,  
    int orientation,  
    int level,  
    const std::string & teamName = "" ) [inline]
```

Definition at line 17 of file PlayerInfoData.hpp.

```
18 : _id(id), _x(x), _y(y), _orientation(orientation), _level(level), _teamName(teamName) {}
```

13.73.3 Member Function Documentation

13.73.3.1 getId()

```
int PlayerInfoData::getId ( ) const [inline]
```

Definition at line 21 of file PlayerInfoData.hpp.

```
21 { return _id; }
```

13.73.3.2 getLevel()

```
int PlayerInfoData::getLevel ( ) const [inline]
```

Definition at line 25 of file PlayerInfoData.hpp.

```
25 { return _level; }
```

13.73.3.3 `getOrientation()`

```
int PlayerInfoData::getOrientation ( ) const [inline]
Definition at line 24 of file PlayerInfoData.hpp.
24 { return _orientation; }
```

13.73.3.4 `getTeamName()`

```
const std::string& PlayerInfoData::getTeamName ( ) const [inline]
Definition at line 26 of file PlayerInfoData.hpp.
26 { return _teamName; }
```

13.73.3.5 `getType()`

```
MessageType PlayerInfoData::getType ( ) const [inline], [override], [virtual]
Implements IMessageData.
Definition at line 19 of file PlayerInfoData.hpp.
19 { return MessageType::PlayerInfo; }
```

13.73.3.6 `getX()`

```
int PlayerInfoData::getX ( ) const [inline]
Definition at line 22 of file PlayerInfoData.hpp.
22 { return _x; }
```

13.73.3.7 `getY()`

```
int PlayerInfoData::getY ( ) const [inline]
Definition at line 23 of file PlayerInfoData.hpp.
23 { return _y; }
```

13.73.3.8 `setId()`

```
void PlayerInfoData::setId (
    int value ) [inline]
Definition at line 28 of file PlayerInfoData.hpp.
28 { _id = value; }
```

13.73.3.9 `setLevel()`

```
void PlayerInfoData::setLevel (
    int value ) [inline]
Definition at line 32 of file PlayerInfoData.hpp.
32 { _level = value; }
```

13.73.3.10 `setOrientation()`

```
void PlayerInfoData::setOrientation (
    int value ) [inline]
Definition at line 31 of file PlayerInfoData.hpp.
31 { _orientation = value; }
```

13.73.3.11 setTeamName()

```
void PlayerInfoData::setTeamName (
    const std::string & value ) [inline]
Definition at line 33 of file PlayerInfoData.hpp.
33 { _teamName = value; }
```

13.73.3.12 setX()

```
void PlayerInfoData::setX (
    int value ) [inline]
Definition at line 29 of file PlayerInfoData.hpp.
29 { _x = value; }
```

13.73.3.13 setY()

```
void PlayerInfoData::setY (
    int value ) [inline]
Definition at line 30 of file PlayerInfoData.hpp.
30 { _y = value; }
```

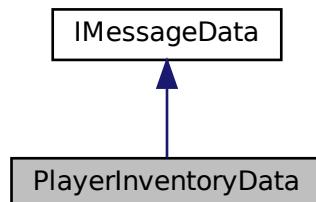
The documentation for this class was generated from the following file:

- /root/Desktop/Zappy/src/GUI/network/protocol/messageData/[PlayerInfoData.hpp](#)

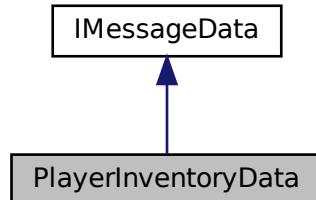
13.74 PlayerInventoryData Class Reference

```
#include <PlayerInventoryData.hpp>
```

Inheritance diagram for PlayerInventoryData:



Collaboration diagram for PlayerInventoryData:



Public Member Functions

- `PlayerInventoryData ()`
- `PlayerInventoryData (int id, int x, int y, int food, int linemate, int deraumere, int sibur, int mendiane, int phiras, int thystame)`
- `MessageType getType () const override`
- `int getId () const`
- `int getFood () const`
- `int getLinemate () const`
- `int getDeraumere () const`
- `int getSibur () const`
- `int getMendiane () const`
- `int getPhiras () const`
- `int getThystame () const`
- `int getX () const`
- `int getY () const`
- `void setId (int value)`
- `void setFood (int value)`
- `void setLinemate (int value)`
- `void setDeraumere (int value)`
- `void setSibur (int value)`
- `void setMendiane (int value)`
- `void setPhiras (int value)`
- `void setThystame (int value)`
- `void setX (int value)`
- `void setY (int value)`

13.74.1 Detailed Description

Definition at line 13 of file PlayerInventoryData.hpp.

13.74.2 Constructor & Destructor Documentation

13.74.2.1 PlayerInventoryData() [1/2]

```
PlayerInventoryData::PlayerInventoryData ( ) [inline]
```

Definition at line 15 of file PlayerInventoryData.hpp.

```
15     : _id(0), _x(0), _y(0), _food(0), _linemate(0), _deraumere(0),
16     _sibur(0), _mendiane(0), _phiras(0), _thystame(0) {}
```

13.74.2.2 PlayerInventoryData() [2/2]

```
PlayerInventoryData::PlayerInventoryData (
    int id,
    int x,
    int y,
    int food,
    int linemate,
    int deraumere,
    int sibur,
    int mendiane,
    int phiras,
    int thystame ) [inline]
Definition at line 17 of file PlayerInventoryData.hpp.
19     : _id(id), _x(x), _y(y), _food(food), _linemate(linemate), _deraumere(deraumere),
20     _sibur(sibur), _mendiane(mendiane), _phiras(phiras), _thystame(thystame) {}
```

13.74.3 Member Function Documentation

13.74.3.1 getDeraumere()

```
int PlayerInventoryData::getDeraumere () const [inline]
Definition at line 26 of file PlayerInventoryData.hpp.
26 { return _deraumere; }
```

13.74.3.2 getFood()

```
int PlayerInventoryData::getFood () const [inline]
Definition at line 24 of file PlayerInventoryData.hpp.
24 { return _food; }
```

13.74.3.3 getId()

```
int PlayerInventoryData::getId () const [inline]
Definition at line 23 of file PlayerInventoryData.hpp.
23 { return _id; }
```

13.74.3.4 getLinemate()

```
int PlayerInventoryData::getLinemate () const [inline]
Definition at line 25 of file PlayerInventoryData.hpp.
25 { return _linemate; }
```

13.74.3.5 getMendiane()

```
int PlayerInventoryData::getMendiane () const [inline]
Definition at line 28 of file PlayerInventoryData.hpp.
28 { return _mendiane; }
```

13.74.3.6 getPhiras()

```
int PlayerInventoryData::getPhiras () const [inline]
Definition at line 29 of file PlayerInventoryData.hpp.
29 { return _phiras; }
```

13.74.3.7 `getSibur()`

```
int PlayerInventoryData::getSibur () const [inline]
Definition at line 27 of file PlayerInventoryData.hpp.
27 { return _sibur; }
```

13.74.3.8 `getThystame()`

```
int PlayerInventoryData::getThystame () const [inline]
Definition at line 30 of file PlayerInventoryData.hpp.
30 { return _thystame; }
```

13.74.3.9 `getType()`

```
MessageType PlayerInventoryData::getType () const [inline], [override], [virtual]
Implements IMessageData.
Definition at line 21 of file PlayerInventoryData.hpp.
21 { return MessageType::PlayerInventory; }
```

13.74.3.10 `getX()`

```
int PlayerInventoryData::getX () const [inline]
Definition at line 31 of file PlayerInventoryData.hpp.
31 { return _x; }
```

13.74.3.11 `getY()`

```
int PlayerInventoryData::getY () const [inline]
Definition at line 32 of file PlayerInventoryData.hpp.
32 { return _y; }
```

13.74.3.12 `setDeraumere()`

```
void PlayerInventoryData::setDeraumere (
    int value ) [inline]
Definition at line 37 of file PlayerInventoryData.hpp.
37 { _deraumere = value; }
```

13.74.3.13 `setFood()`

```
void PlayerInventoryData::setFood (
    int value ) [inline]
Definition at line 35 of file PlayerInventoryData.hpp.
35 { _food = value; }
```

13.74.3.14 `setId()`

```
void PlayerInventoryData::setId (
    int value ) [inline]
Definition at line 34 of file PlayerInventoryData.hpp.
34 { _id = value; }
```

13.74.3.15 setLinemate()

```
void PlayerInventoryData::setLinemate (
    int value ) [inline]
Definition at line 36 of file PlayerInventoryData.hpp.
36 { _linemate = value; }
```

13.74.3.16 setMendiane()

```
void PlayerInventoryData::setMendiane (
    int value ) [inline]
Definition at line 39 of file PlayerInventoryData.hpp.
39 { _mendiane = value; }
```

13.74.3.17 setPhiras()

```
void PlayerInventoryData::setPhiras (
    int value ) [inline]
Definition at line 40 of file PlayerInventoryData.hpp.
40 { _phiras = value; }
```

13.74.3.18 setSibur()

```
void PlayerInventoryData::setSibur (
    int value ) [inline]
Definition at line 38 of file PlayerInventoryData.hpp.
38 { _sibur = value; }
```

13.74.3.19 setThystame()

```
void PlayerInventoryData::setThystame (
    int value ) [inline]
Definition at line 41 of file PlayerInventoryData.hpp.
41 { _thystame = value; }
```

13.74.3.20 setX()

```
void PlayerInventoryData::setX (
    int value ) [inline]
Definition at line 42 of file PlayerInventoryData.hpp.
42 { _x = value; }
```

13.74.3.21 setY()

```
void PlayerInventoryData::setY (
    int value ) [inline]
Definition at line 43 of file PlayerInventoryData.hpp.
43 { _y = value; }
```

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

- /root/Desktop/Zappy/src/GUI/network/protocol/messageData/PlayerInventoryData.hpp

13.75 poll_manager_s Struct Reference

Structure de gestion du polling non-bloquant.

```
#include <server.h>
```

Data Fields

- struct pollfd * **fds**
Tableau des descripteurs de fichiers.
- int **capacity**
Capacité du tableau.
- bool **needs_rebuild**
Flag indiquant si une reconstruction est nécessaire.

13.75.1 Detailed Description

Structure de gestion du polling non-bloquant.

Cette structure gère les descripteurs de fichiers pour le polling et optimise les performances en évitant les reconstructions inutiles.

Definition at line 23 of file server.h.

13.75.2 Field Documentation

13.75.2.1 capacity

```
int poll_manager_s::capacity
```

Capacité du tableau.

Definition at line 25 of file server.h.

13.75.2.2 fds

```
struct pollfd* poll_manager_s::fds
```

Tableau des descripteurs de fichiers.

Definition at line 24 of file server.h.

13.75.2.3 needs_rebuild

```
bool poll_manager_s::needs_rebuild
```

Flag indiquant si une reconstruction est nécessaire.

Definition at line 26 of file server.h.

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

- /root/Desktop/Zappy/src/Server/include/server.h

13.76 RayGUICPP::ProgressBar Class Reference

```
#include <RayGuiEncap.hpp>
```

Static Public Member Functions

- static float **Draw** (const Rectangle &bounds, const std::string &textLeft, const std::string &textRight, float value, float minValue, float maxValue)

13.76.1 Detailed Description

Definition at line 143 of file RayGuiEncap.hpp.

13.76.2 Member Function Documentation

13.76.2.1 Draw()

```
static float RayGUICPP::ProgressBar::Draw (
    const Rectangle & bounds,
    const std::string & textLeft,
    const std::string & textRight,
    float value,
    float minValue,
    float maxValue ) [inline], [static]
Definition at line 145 of file RayGuiEncap.hpp.
145
146     float v = value;
147     GuiProgressBar(bounds, textLeft.c_str(), textRight.c_str(), &v, minValue, maxValue);
148     return v;
149 }
```

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

- /root/Desktop/Zappy/libs/RayGUICPP/include/RayGuiEncap.hpp

13.77 ProtocolParser Class Reference

```
#include <ProtocolParser.hpp>
```

Public Types

- `typedef std::function< Message(const std::string &) > ParseFunction`

Public Member Functions

- `ProtocolParser ()`
- `~ProtocolParser ()=default`
- `Message parseMessage (const std::string &message)`
- `std::pair< std::string, std::string > parseMessage (const std::string &message) const`
- `std::string getCommandName (const std::string &message)`
- `bool messageComplete (const std::string &buffer)`
- `std::vector< std::string > splitMessage (const std::string &message)`
- `std::string extractCommandParameter (const std::string &message)`
- `std::vector< std::string > extractMessageParameters (const std::string &message)`
- `int parseIntParameter (const std::string ¶m)`
- `void handleProtocol (const std::string &protocol)`
- `Message parseMapSize (const std::string &message)`
- `Message parseTileContent (const std::string &message)`
- `Message parseAllTilesContent (const std::string &message)`
- `Message parseTeamNames (const std::string &message)`
- `Message parsePlayerConnection (const std::string &message)`
- `Message parsePlayerPosition (const std::string &message)`
- `Message parsePlayerLevel (const std::string &message)`
- `Message parsePlayerInventory (const std::string &message)`
- `Message parsePlayerExpulsion (const std::string &message)`
- `Message parsePlayerBroadcast (const std::string &message)`
- `Message parsePlayerDeath (const std::string &message)`
- `Message parseResourceDrop (const std::string &message)`
- `Message parseResourceCollect (const std::string &message)`
- `Message parseIncantationStart (const std::string &message)`

- `Message parseIncantationEnd (const std::string &message)`
- `Message parseEggDrop (const std::string &message)`
- `Message parseEggConnection (const std::string &message)`
- `Message parseEggDeath (const std::string &message)`
- `Message parseEggLaying (const std::string &message)`
- `Message parseTimeUnit (const std::string &message)`
- `Message parseEndGame (const std::string &message)`
- `Message parseServerMessage (const std::string &message)`
- `Message parseUnknownCommand (const std::string &message)`

13.77.1 Detailed Description

Definition at line 22 of file ProtocolParser.hpp.

13.77.2 Member Typedef Documentation

13.77.2.1 ParseFunction

```
typedef std::function<Message(const std::string&)> ProtocolParser::ParseFunction
```

Definition at line 24 of file ProtocolParser.hpp.

13.77.3 Constructor & Destructor Documentation

13.77.3.1 ProtocolParser()

```
ProtocolParser::ProtocolParser ()
```

Definition at line 19 of file ProtocolParser.cpp.

```
19
20     _headerHandlers = std::map<std::string, ParseFunction>{
21         {MSZ_HEADER, [this](const std::string& msg) { return this->parseMapSize(msg); }},
22         {BCT_HEADER, [this](const std::string& msg) { return this->parseTileContent(msg); }},
23         {MCT_HEADER, [this](const std::string& msg) { return this->parseAllTilesContent(msg); }},
24         {TNA_HEADER, [this](const std::string& msg) { return this->parseTeamNames(msg); }},
25         {PNW_HEADER, [this](const std::string& msg) { return this->parsePlayerConnection(msg); }},
26         {PPO_HEADER, [this](const std::string& msg) { return this->parsePlayerPosition(msg); }},
27         {PLV_HEADER, [this](const std::string& msg) { return this->parsePlayerLevel(msg); }},
28         {PIN_HEADER, [this](const std::string& msg) { return this->parsePlayerInventory(msg); }},
29         {PEX_HEADER, [this](const std::string& msg) { return this->parsePlayerExpulsion(msg); }},
30         {PBC_HEADER, [this](const std::string& msg) { return this->parsePlayerBroadcast(msg); }},
31         {PDR_HEADER, [this](const std::string& msg) { return this->parseRessourceDrop(msg); }},
32         {PGT_HEADER, [this](const std::string& msg) { return this->parseRessourceCollect(msg); }},
33         {PDI_HEADER, [this](const std::string& msg) { return this->parsePlayerDeath(msg); }},
34         {PIC_HEADER, [this](const std::string& msg) { return this->parseIncantationStart(msg); }},
35         {PIE_HEADER, [this](const std::string& msg) { return this->parseIncantationEnd(msg); }},
36         {ENW_HEADER, [this](const std::string& msg) { return this->parseEggDrop(msg); }},
37         {EBO_HEADER, [this](const std::string& msg) { return this->parseEggConnection(msg); }},
38         {EDI_HEADER, [this](const std::string& msg) { return this->parseEggDeath(msg); }},
39         {PFK_HEADER, [this](const std::string& msg) { return this->parseEggLaying(msg); }},
40         {SGT_HEADER, [this](const std::string& msg) { return this->parseTimeUnit(msg); }},
41         {SST_HEADER, [this](const std::string& msg) { return this->parseTimeUnit(msg); }},
42         {SEG_HEADER, [this](const std::string& msg) { return this->parseEndGame(msg); }},
43         {SMG_HEADER, [this](const std::string& msg) { return this->parseServerMessage(msg); }},
44         {SUC_HEADER, [this](const std::string& msg) { return this->parseUnknownCommand(msg); }},
45         {SBP_HEADER, [this](const std::string& msg) { return this->parseUnknownCommand(msg); }}
46     };
47 }
```

13.77.3.2 ~ProtocolParser()

```
ProtocolParser::~ProtocolParser () [default]
```

13.77.4 Member Function Documentation

13.77.4.1 extractCommandParameter()

```
std::string ProtocolParser::extractCommandParameter (
    const std::string & message )
```

Definition at line 115 of file ProtocolParser.cpp.

```
115
116     size_t spacePos = message.find(' ');
117     if (spacePos == std::string::npos)
118         return "";
119     size_t newlinePos = message.find('\n', spacePos);
120     if (newlinePos == std::string::npos)
121         newlinePos = message.length();
122     return message.substr(spacePos + 1, newlinePos - spacePos - 1);
123 }
```

13.77.4.2 extractMessageParameters()

```
std::vector< std::string > ProtocolParser::extractMessageParameters (
    const std::string & message )
```

Definition at line 108 of file ProtocolParser.cpp.

```
108
109     auto parts = splitMessage(message);
110     if (!parts.empty())
111         parts.erase(parts.begin());
112     return parts;
113 }
```

13.77.4.3 getCommandName()

```
std::string ProtocolParser::getCommandName (
    const std::string & message )
```

Definition at line 87 of file ProtocolParser.cpp.

```
87
88     size_t spacePos = message.find(' ');
89     size_t newlinePos = message.find('\n');
90     size_t endPos = std::min(spacePos, newlinePos);
91     if (endPos == std::string::npos)
92         endPos = message.length();
93     return message.substr(0, endPos);
94 }
```

13.77.4.4 handleProtocol()

```
void ProtocolParser::handleProtocol (
    const std::string & protocol )
```

13.77.4.5 messageComplete()

```
bool ProtocolParser::messageComplete (
    const std::string & buffer )
```

Definition at line 83 of file ProtocolParser.cpp.

```
83
84     return buffer.find('\n') != std::string::npos;
85 }
```

13.77.4.6 parseAllTilesContent()

```
Message ProtocolParser::parseAllTilesContent (
    const std::string & message )
```

Definition at line 159 of file ProtocolParser.cpp.

```
159
160     std::vector<std::string> params = extractMessageParameters(message);
161     if (!params.empty())
162         throw ProtocolParserException("Invalid parameters for all tiles content: " + message);
163     return Message(MCT_HEADER, extractCommandParameter(message), nullptr);
164 }
```

13.77.4.7 parseEggConnection()

```
Message ProtocolParser::parseEggConnection (
    const std::string & message )
```

Definition at line 338 of file ProtocolParser.cpp.

```
338
339     std::vector<std::string> params = extractMessageParameters(message);
340     if (params.size() < 1)
341         throw ProtocolParserException("Invalid egg connection parameters: " + message);
342     int eggId = parseIntParameter(params[0]);
343
344     auto eggData = std::make_shared<EggData>(eggId, 0, 0, 0, EggData::EggAction::Connection);
345     return Message(EBO_HEADER, extractCommandParameter(message), eggData);
346 }
```

13.77.4.8 parseEggDeath()

```
Message ProtocolParser::parseEggDeath (
    const std::string & message )
```

Definition at line 348 of file ProtocolParser.cpp.

```
348
349     std::vector<std::string> params = extractMessageParameters(message);
350     if (params.size() < 1)
351         throw ProtocolParserException("Invalid egg death parameters: " + message);
352     int eggId = parseIntParameter(params[0]);
353
354     auto eggData = std::make_shared<EggData>(eggId, 0, 0, 0, EggData::EggAction::Death);
355     return Message(EDI_HEADER, extractCommandParameter(message), eggData);
356 }
```

13.77.4.9 parseEggDrop()

```
Message ProtocolParser::parseEggDrop (
    const std::string & message )
```

Definition at line 325 of file ProtocolParser.cpp.

```
325
326     std::vector<std::string> params = extractMessageParameters(message);
327     if (params.size() < 4)
328         throw ProtocolParserException("Invalid egg drop parameters: " + message);
329     int eggId = parseIntParameter(params[0]);
330     int playerId = parseIntParameter(params[1]);
331     int x = parseIntParameter(params[2]);
332     int y = parseIntParameter(params[3]);
333
334     auto eggData = std::make_shared<EggData>(eggId, playerId, x, y, EggData::EggAction::Drop);
335     return Message(ENW_HEADER, extractCommandParameter(message), eggData);
336 }
```

13.77.4.10 parseEggLaying()

```
Message ProtocolParser::parseEggLaying (
    const std::string & message )
```

Definition at line 315 of file ProtocolParser.cpp.

```
315
316     std::vector<std::string> params = extractMessageParameters(message);
317     if (params.size() < 1)
318         throw ProtocolParserException("Invalid egg laying parameters: " + message);
319     int playerId = parseIntParameter(params[0]);
320
321     auto eggData = std::make_shared<EggData>(0, playerId, 0, 0, EggData::EggAction::Laying);
322     return Message(PFK_HEADER, extractCommandParameter(message), eggData);
323 }
```

13.77.4.11 parseEndGame()

```
Message ProtocolParser::parseEndGame (
    const std::string & message )
```

Definition at line 369 of file ProtocolParser.cpp.

```
369
370     std::vector<std::string> params = extractMessageParameters(message);
371     if (params.size() < 1)
372         throw ProtocolParserException("Invalid end game parameters: " + message);
373     std::string teamName = params[0];
374
375     auto endGameData = std::make_shared<EndGameData>(teamName);
376     return Message(SEG_HEADER, extractCommandParameter(message), endGameData);
377 }
```

13.77.4.12 parseIncantationEnd()

```
Message ProtocolParser::parseIncantationEnd (
    const std::string & message )
```

Definition at line 301 of file ProtocolParser.cpp.

```
301
302     std::vector<std::string> params = extractMessageParameters(message);
303     if (params.size() < 3)
304         throw ProtocolParserException("Invalid incantation end parameters: " + message);
305
306     int x = parseIntParameter(params[0]);
307     int y = parseIntParameter(params[1]);
308     bool result = parseIntParameter(params[2]);
309
310     auto incantationEndData = std::make_shared<IncantationEndData>(x, y, result);
311     return Message(PIE_HEADER, extractCommandParameter(message), incantationEndData);
312 }
```

13.77.4.13 parseIncantationStart()

```
Message ProtocolParser::parseIncantationStart (
    const std::string & message )
```

Definition at line 285 of file ProtocolParser.cpp.

```
285
286     std::vector<std::string> params = extractMessageParameters(message);
287     if (params.size() < 4)
288         throw ProtocolParserException("Invalid incantation start parameters: " + message);
289
290     int x = parseIntParameter(params[0]);
291     int y = parseIntParameter(params[1]);
292     int level = parseIntParameter(params[2]);
293     std::vector<int> playerIds;
294
295     for (size_t i = 3; i < params.size(); i++)
296         playerIds.push_back(parseIntParameter(params[i]));
297     auto incantationData = std::make_shared<IncantationData>(x, y, level, playerIds);
298     return Message(PIC_HEADER, extractCommandParameter(message), incantationData);
299 }
```

13.77.4.14 parseIntParameter()

```
int ProtocolParser::parseIntParameter (
    const std::string & param )
```

Definition at line 125 of file ProtocolParser.cpp.

```
125
126     try {
127         return std::stoi(param);
128     } catch (const std::exception &e) {
129         throw ProtocolParserException("Failed to parse integer parameter: " + param);
130     }
131 }
```

13.77.4.15 parseMapSize()

```
Message ProtocolParser::parseMapSize (
    const std::string & message )
```

Definition at line 133 of file ProtocolParser.cpp.

```
133
134     std::vector<std::string> params = extractMessageParameters(message);
135     if (params.size() != 2)
136         throw ProtocolParserException("Invalid map size parameters: " + message);
137     auto mapSizeData = std::make_shared<MapSizeData>(parseIntParameter(params[0]),
138             parseIntParameter(params[1]));
139     return Message(MSZ_HEADER, extractCommandParameter(message), mapSizeData);
```

13.77.4.16 parseMessage() [1/2]

```
Message ProtocolParser::parseMessage (
    const std::string & message )
```

Definition at line 49 of file ProtocolParser.cpp.

```
49
50     if (message.empty())
51         throw ProtocolParserException("Empty message received");
52     if (!messageComplete(message))
53         throw ProtocolParserException("Incomplete message received: " + message);
54     std::string header = getCommandName(message);
55     auto handlerIt = _headerHandlers.find(header);
56     if (handlerIt != _headerHandlers.end())
57         return handlerIt->second(message);
58     return parseUnknownCommand(message);
59 }
```

13.77.4.17 parseMessage() [2/2]

```
std::pair< std::string, std::string > ProtocolParser::parseMessage (
    const std::string & message ) const
```

Definition at line 61 of file ProtocolParser.cpp.

```
61
62     std::string header;
63     std::string data;
64     size_t spacePos = message.find(' ');
65     size_t newlinePos = message.find('\n');
66
67     if (spacePos != std::string::npos && (newlinePos == std::string::npos || spacePos < newlinePos)) {
68         header = message.substr(0, spacePos);
69         if (newlinePos != std::string::npos)
70             data = message.substr(spacePos + 1, newlinePos - spacePos - 1);
71     } else
72         data = message.substr(spacePos + 1);
73     } else if (newlinePos != std::string::npos) {
74         header = message.substr(0, newlinePos);
75         data = "";
76     } else {
77         header = message;
78         data = "";
79     }
80     return std::make_pair(header, data);
81 }
```

13.77.4.18 parsePlayerBroadcast()

```
Message ProtocolParser::parsePlayerBroadcast (
    const std::string & message )
```

Definition at line 242 of file ProtocolParser.cpp.

```
242
243     std::vector<std::string> params = extractMessageParameters(message);
244     if (params.size() < 2)
245         throw ProtocolParserException("Invalid player broadcast parameters: " + message);
246     int id = parseIntParameter(params[0]);
247     std::string broadcastMessage = params[1];
248
249     auto broadcastData = std::make_shared<BroadcastData>(id, broadcastMessage);
250     return Message(PBC_HEADER, extractCommandParameter(message), broadcastData);
251 }
```

13.77.4.19 parsePlayerConnection()

```
Message ProtocolParser::parsePlayerConnection (
    const std::string & message )
```

Definition at line 174 of file ProtocolParser.cpp.

```
174
175     std::vector<std::string> params = extractMessageParameters(message);
176     if (params.size() < 6)
177         throw ProtocolParserException("Invalid player connection parameters: " + message);
178     int id = parseIntParameter(params[0]);
179     int x = parseIntParameter(params[1]);
180     int y = parseIntParameter(params[2]);
181     int orientation = parseIntParameter(params[3]);
182     int level = parseIntParameter(params[4]);
183     std::string teamName = params[5];
184
185     auto playerInfoData = std::make_shared<PlayerInfoData>(id, x, y, orientation, level, teamName);
186     return Message(PNW_HEADER, extractCommandParameter(message), playerInfoData);
187 }
```

13.77.4.20 parsePlayerDeath()

```
Message ProtocolParser::parsePlayerDeath (
    const std::string & message )
```

Definition at line 253 of file ProtocolParser.cpp.

```
253
254     std::vector<std::string> params = extractMessageParameters(message);
255     if (params.size() < 1)
256         throw ProtocolParserException("Invalid player death parameters: " + message);
257     int id = parseIntParameter(params[0]);
258
259     auto playerInfoData = std::make_shared<PlayerInfoData>(id, 0, 0, 0, 0);
260     return Message(PDI_HEADER, extractCommandParameter(message), playerInfoData);
261 }
```

13.77.4.21 parsePlayerExpulsion()

```
Message ProtocolParser::parsePlayerExpulsion (
    const std::string & message )
```

Definition at line 232 of file ProtocolParser.cpp.

```
232
233     std::vector<std::string> params = extractMessageParameters(message);
234     if (params.size() < 1)
235         throw ProtocolParserException("Invalid player expulsion parameters: " + message);
236     int id = parseIntParameter(params[0]);
237
238     auto playerInfoData = std::make_shared<PlayerInfoData>(id, 0, 0, 0, 0);
239     return Message(PEX_HEADER, extractCommandParameter(message), playerInfoData);
240 }
```

13.77.4.22 parsePlayerInventory()

```
Message ProtocolParser::parsePlayerInventory (
    const std::string & message )
```

Definition at line 213 of file ProtocolParser.cpp.

```
213
214     std::vector<std::string> params = extractMessageParameters(message);
215     if (params.size() < 10)
216         throw ProtocolParserException("Invalid player inventory parameters: " + message);
217     int id = parseIntParameter(params[0]);
218     int x = parseIntParameter(params[1]);
219     int y = parseIntParameter(params[2]);
220     int food = parseIntParameter(params[3]);
221     int linemate = parseIntParameter(params[4]);
222     int deraumere = parseIntParameter(params[5]);
223     int sibur = parseIntParameter(params[6]);
224     int mendiane = parseIntParameter(params[7]);
225     int phiras = parseIntParameter(params[8]);
226     int thystame = parseIntParameter(params[9]);
227 }
```

```

228     auto inventoryData = std::make_shared<PlayerInventoryData>(id, x, y, food, linemate, deraumere,
229     sibur, mendiane, phiras, thystame);
230 }  


```

13.77.4.23 parsePlayerLevel()

```
Message ProtocolParser::parsePlayerLevel (
    const std::string & message )
```

Definition at line 202 of file ProtocolParser.cpp.

```

202
203     std::vector<std::string> params = extractMessageParameters(message);
204     if (params.size() < 2)
205         throw ProtocolParserException("Invalid player level parameters: " + message);
206     int id = parseIntParameter(params[0]);
207     int level = parseIntParameter(params[1]);
208
209     auto playerInfoData = std::make_shared<PlayerInfoData>(id, 0, 0, 0, level);
210     return Message(PLV_HEADER, extractCommandParameter(message), playerInfoData);
211 }
```

13.77.4.24 parsePlayerPosition()

```
Message ProtocolParser::parsePlayerPosition (
    const std::string & message )
```

Definition at line 189 of file ProtocolParser.cpp.

```

189
190     std::vector<std::string> params = extractMessageParameters(message);
191     if (params.size() < 4)
192         throw ProtocolParserException("Invalid player position parameters: " + message);
193     int id = parseIntParameter(params[0]);
194     int x = parseIntParameter(params[1]);
195     int y = parseIntParameter(params[2]);
196     int orientation = parseIntParameter(params[3]);
197
198     auto playerInfoData = std::make_shared<PlayerInfoData>(id, x, y, orientation, 0);
199     return Message(PPO_HEADER, extractCommandParameter(message), playerInfoData);
200 }
```

13.77.4.25 parseRessourceCollect()

```
Message ProtocolParser::parseRessourceCollect (
    const std::string & message )
```

Definition at line 274 of file ProtocolParser.cpp.

```

274
275     std::vector<std::string> params = extractMessageParameters(message);
276     if (params.size() < 2)
277         throw ProtocolParserException("Invalid resource collection parameters: " + message);
278     int playerId = parseIntParameter(params[0]);
279     int resourceType = parseIntParameter(params[1]);
280
281     auto resourceData = std::make_shared<ResourceData>(playerId, resourceType);
282     return Message(PGT_HEADER, extractCommandParameter(message), resourceData);
283 }
```

13.77.4.26 parseRessourceDrop()

```
Message ProtocolParser::parseRessourceDrop (
    const std::string & message )
```

Definition at line 263 of file ProtocolParser.cpp.

```

263
264     std::vector<std::string> params = extractMessageParameters(message);
265     if (params.size() < 2)
266         throw ProtocolParserException("Invalid resource drop parameters: " + message);
267     int playerId = parseIntParameter(params[0]);
268     int resourceType = parseIntParameter(params[1]);
269
270     auto resourceData = std::make_shared<ResourceData>(playerId, resourceType);
271     return Message(PDR_HEADER, extractCommandParameter(message), resourceData);
272 }
```

13.77.4.27 parseServerMessage()

```
Message ProtocolParser::parseServerMessage (
    const std::string & message )
```

Definition at line 379 of file ProtocolParser.cpp.

```
379
380     std::vector<std::string> params = extractMessageParameters(message);
381     if (params.empty())
382         throw ProtocolParserException("Invalid server message parameters: " + message);
383     std::string serverMessage = params[0];
384
385     auto serverMessageData = std::make_shared<ServerMessageData>(serverMessage);
386     return Message(SMG_HEADER, extractCommandParameter(message), serverMessageData);
387 }
```

13.77.4.28 parseTeamNames()

```
Message ProtocolParser::parseTeamNames (
    const std::string & message )
```

Definition at line 166 of file ProtocolParser.cpp.

```
166
167     std::vector<std::string> params = extractMessageParameters(message);
168     if (params.size() < 1)
169         throw ProtocolParserException("Invalid parameters for team names: " + message);
170     auto teamNamesData = std::make_shared<TeamNameData>(params);
171     return Message(TNA_HEADER, extractCommandParameter(message), teamNamesData);
172 }
```

13.77.4.29 parseTileContent()

```
Message ProtocolParser::parseTileContent (
    const std::string & message )
```

Definition at line 141 of file ProtocolParser.cpp.

```
141
142     std::vector<std::string> params = extractMessageParameters(message);
143     if (params.size() < 8)
144         throw ProtocolParserException("Invalid tile content parameters: " + message);
145     int x = parseIntParameter(params[0]);
146     int y = parseIntParameter(params[1]);
147     int food = parseIntParameter(params[2]);
148     int linemate = parseIntParameter(params[3]);
149     int deraumere = parseIntParameter(params[4]);
150     int sibur = parseIntParameter(params[5]);
151     int mendiane = parseIntParameter(params[6]);
152     int phiras = parseIntParameter(params[7]);
153     int thystame = parseIntParameter(params[8]);
154
155     auto tileContentData = std::make_shared<TileContentData>(x, y, food, linemate, deraumere, sibur,
156     mendiane, phiras, thystame);
156     return Message(BCT_HEADER, extractCommandParameter(message), tileContentData);
157 }
```

13.77.4.30 parseTimeUnit()

```
Message ProtocolParser::parseTimeUnit (
    const std::string & message )
```

Definition at line 358 of file ProtocolParser.cpp.

```
358
359     std::vector<std::string> params = extractMessageParameters(message);
360     if (params.size() < 1)
361         throw ProtocolParserException("Invalid time unit parameters: " + message);
362     int timeUnit = parseIntParameter(params[0]);
363     std::string header = getCommandName(message);
364
365     auto timeUnitData = std::make_shared<TimeUnitData>(timeUnit);
366     return Message(header, extractCommandParameter(message), timeUnitData);
367 }
```

13.77.4.31 parseUnknownCommand()

```
Message ProtocolParser::parseUnknownCommand (
    const std::string & message )
```

Definition at line 389 of file ProtocolParser.cpp.

```
389
390     std::cout << "Unknown command received: " << message << std::endl;
391
392     auto serverMessageData = std::make_shared<ServerMessageData>("Unknown command: " + message);
393     return Message(SUC_HEADER, "", serverMessageData);
394 }
```

13.77.4.32 splitMessage()

```
std::vector< std::string > ProtocolParser::splitMessage (
    const std::string & message )
```

Definition at line 96 of file ProtocolParser.cpp.

```
96
97     std::vector<std::string> parts;
98     std::string cleanMessage = message;
99     if (!cleanMessage.empty() && cleanMessage.back() == '\n')
100         cleanMessage.pop_back();
101     std::istringstream ss(cleanMessage);
102     std::string token;
103     while (ss >> token)
104         parts.push_back(token);
105     return parts;
106 }
```

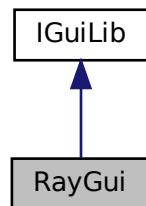
The documentation for this class was generated from the following files:

- /root/Desktop/Zappy/src/GUI/network/protocol/ProtocolParser.hpp
- /root/Desktop/Zappy/src/GUI/network/protocol/ProtocolParser.cpp

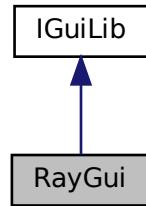
13.78 RayGui Class Reference

```
#include <RayGui.hpp>
```

Inheritance diagram for RayGui:



Collaboration diagram for RayGui:



Public Member Functions

- [RayGui \(\)](#)
- [~RayGui \(\) override](#)
- void [DrawButton](#) (float x, float y, float width, float height, const std::string &text) override
- bool [ButtonPressed](#) (float x, float y, float width, float height, const std::string &text) override
- void [DrawLabel](#) (float x, float y, float width, float height, const std::string &text) override
- bool [DrawCheckBox](#) (float x, float y, float width, float height, const std::string &text, bool checked) override
- bool [DrawToggle](#) (float x, float y, float width, float height, const std::string &text, bool active) override
- int [DrawToggleGroup](#) (float x, float y, float width, float height, const std::string &text, int active) override
- int [DrawComboBox](#) (float x, float y, float width, float height, const std::string &text, int active) override
- int [DrawDropdownBox](#) (float x, float y, float width, float height, const std::string &text, int active, bool editMode) override
- bool [DrawTextBox](#) (float x, float y, float width, float height, std::shared_ptr< std::string > text, int maxLength, bool editMode) override
- bool [DrawValueBox](#) (float x, float y, float width, float height, std::shared_ptr< int > value, int minValue, int maxValue, bool editMode) override
- bool [DrawSpinner](#) (float x, float y, float width, float height, std::shared_ptr< int > value, int minValue, int maxValue, bool editMode) override
- float [DrawSlider](#) (float x, float y, float width, float height, const std::string &textLeft, const std::string &textRight, float value, float minValue, float maxValue) override
- float [DrawProgressBar](#) (float x, float y, float width, float height, const std::string &textLeft, const std::string &textRight, float value, float minValue, float maxValue) override
- void [DrawStatusBar](#) (float x, float y, float width, float height, const std::string &text) override
- bool [DrawWindowBox](#) (float x, float y, float width, float height, const std::string &title) override
- void [DrawGroupBox](#) (float x, float y, float width, float height, const std::string &text) override
- void [DrawPanel](#) (float x, float y, float width, float height) override
- [ZappyTypes::Rectangle DrawScrollPane](#) (float x, float y, float width, float height, const std::string &text, [ZappyTypes::Rectangle](#) content, std::shared_ptr< [ZappyTypes::Vector2](#) > scroll) override
- int [DrawTabBar](#) (float x, float y, float width, float height, const std::vector< std::string > &tabs, int active) override
- int [DrawListView](#) (float x, float y, float width, float height, const std::vector< std::string > &items, int active) override
- [ZappyTypes::Color DrawColorPicker](#) (float x, float y, float width, float height, [ZappyTypes::Color](#) color) override
- int [DrawMessageBox](#) (float x, float y, float width, float height, const std::string &title, const std::string &message, const std::string &buttons) override
- int [DrawTextInputBox](#) (float x, float y, float width, float height, const std::string &title, const std::string &message, std::shared_ptr< std::string > text, const std::string &buttons) override
- std::string [IconText](#) (int iconId, const std::string &text) override

- void `LoadStyle` (const std::string &file) override
- void `SetStyle` (int control, int property, int value) override
- int `GetStyle` (int control, int property) override

13.78.1 Detailed Description

Definition at line 15 of file RayGui.hpp.

13.78.2 Constructor & Destructor Documentation

13.78.2.1 RayGui()

```
RayGui::RayGui ( )
Definition at line 14 of file RayGui.cpp.
14 { }
```

13.78.2.2 ~RayGui()

```
RayGui::~RayGui ( ) [override]
Definition at line 15 of file RayGui.cpp.
15 { }
```

13.78.3 Member Function Documentation

13.78.3.1 ButtonPressed()

```
bool RayGui::ButtonPressed (
    float x,
    float y,
    float width,
    float height,
    const std::string & text ) [override], [virtual]
Implements IGuiLib.
Definition at line 20 of file RayGui.cpp.
20
21     return RayGUICPP::Button::Draw({x, y, width, height}, text);
22 }
```

13.78.3.2 DrawButton()

```
void RayGui::DrawButton (
    float x,
    float y,
    float width,
    float height,
    const std::string & text ) [override], [virtual]
Implements IGuiLib.
Definition at line 17 of file RayGui.cpp.
17
18     RayGUICPP::Button::Draw({x, y, width, height}, text);
19 }
```

13.78.3.3 DrawCheckBox()

```
bool RayGui::DrawCheckBox (
    float x,
    float y,
    float width,
    float height,
    const std::string & text,
    bool checked ) [override], [virtual]
```

Implements [IGuiLib](#).

Definition at line 26 of file RayGui.cpp.

```
26
27     {
28         return RayGUICPP::CheckBox::Draw({x, y, width, height}, text, checked);
29     }
```

13.78.3.4 DrawColorPicker()

```
ZappyTypes::Color RayGui::DrawColorPicker (
    float x,
    float y,
    float width,
    float height,
    ZappyTypes::Color color ) [override], [virtual]
```

Implements [IGuiLib](#).

Definition at line 82 of file RayGui.cpp.

```
82
83     {
84         return TypeAdapter::FromRaylib(RayGUICPP::ColorPicker::Draw({x, y, width, height},
85             TypeAdapter::ToRaylib(color)));
86     }
```

13.78.3.5 DrawComboBox()

```
int RayGui::DrawComboBox (
    float x,
    float y,
    float width,
    float height,
    const std::string & text,
    int active ) [override], [virtual]
```

Implements [IGuiLib](#).

Definition at line 35 of file RayGui.cpp.

```
35
36     return RayGUICPP::ComboBox::Draw({x, y, width, height}, text, active);
37 }
```

{

13.78.3.6 DrawDropdownBox()

```
int RayGui::DrawDropdownBox (
    float x,
    float y,
    float width,
    float height,
    const std::string & text,
    int active,
    bool editMode ) [override], [virtual]
```

Implements [IGuiLib](#).

Definition at line 38 of file RayGui.cpp.

```
38
39     {
```

```
39     return RayGUICPP::DropdownBox::Draw({x, y, width, height}, text, active, editMode);
40 }
```

13.78.3.7 DrawGroupBox()

```
void RayGui::DrawGroupBox (
    float x,
    float y,
    float width,
    float height,
    const std::string & text ) [override], [virtual]
```

Implements [IGuiLib](#).

Definition at line 62 of file RayGui.cpp.

```
62
63     RayGUICPP::GroupBox::Draw({x, y, width, height}, text);
64 }
```

13.78.3.8 DrawLabel()

```
void RayGui::DrawLabel (
    float x,
    float y,
    float width,
    float height,
    const std::string & text ) [override], [virtual]
```

Implements [IGuiLib](#).

Definition at line 23 of file RayGui.cpp.

```
23
24     RayGUICPP::Label::Draw({x, y, width, height}, text);
25 }
```

13.78.3.9 DrawListView()

```
int RayGui::DrawListView (
    float x,
    float y,
    float width,
    float height,
    const std::vector< std::string > & items,
    int active ) [override], [virtual]
```

Implements [IGuiLib](#).

Definition at line 79 of file RayGui.cpp.

```
79
80     {
81         return RayGUICPP::ListView::Draw({x, y, width, height}, items, active);
82     }
```

13.78.3.10 DrawMessageBox()

```
int RayGui::DrawMessageBox (
    float x,
    float y,
    float width,
    float height,
    const std::string & title,
    const std::string & message,
    const std::string & buttons ) [override], [virtual]
```

Implements [IGuiLib](#).

Definition at line 85 of file RayGui.cpp.

```

85
86     return RayGUICPP::MessageBox::Draw({x, y, width, height}, title, message, buttons);
87 }

```

13.78.3.11 DrawPanel()

```

void RayGui::DrawPanel (
    float x,
    float y,
    float width,
    float height ) [override], [virtual]

```

Implements [IGuiLib](#).

Definition at line 65 of file RayGui.cpp.

```

65
66     RayGUICPP::Panel::Draw({x, y, width, height});           {
67 }

```

13.78.3.12 DrawProgressBar()

```

float RayGui::DrawProgressBar (
    float x,
    float y,
    float width,
    float height,
    const std::string & textLeft,
    const std::string & textRight,
    float value,
    float minValue,
    float maxValue ) [override], [virtual]

```

Implements [IGuiLib](#).

Definition at line 53 of file RayGui.cpp.

```

53
54     return RayGUICPP::ProgressBar::Draw({x, y, width, height}, textLeft, textRight, value, minValue,
55                                         maxValue);           {
55 }

```

13.78.3.13 DrawScrollPane()

```

ZappyTypes::Rectangle RayGui::DrawScrollPane (
    float x,
    float y,
    float width,
    float height,
    const std::string & text,
    ZappyTypes::Rectangle content,
    std::shared_ptr< ZappyTypes::Vector2 > scroll ) [override], [virtual]

```

Implements [IGuiLib](#).

Definition at line 68 of file RayGui.cpp.

```

68
69     if (!scroll) return ZappyTypes::Rectangle{};
70     auto raylibContent = TypeAdapter::ToRaylib(content);
71     auto raylibScroll = std::make_shared<Vector2>(TypeAdapter::ToRaylib(*scroll));
72     auto result = RayGUICPP::ScrollPane::Draw({x, y, width, height}, text, raylibContent, raylibScroll);
73     *scroll = TypeAdapter::FromRaylib(*raylibScroll);
74     return TypeAdapter::FromRaylib(result);
75 }

```

13.78.3.14 DrawSlider()

```
float RayGui::DrawSlider (
    float x,
    float y,
    float width,
    float height,
    const std::string & textLeft,
    const std::string & textRight,
    float value,
    float minValue,
    float maxValue ) [override], [virtual]
```

Implements [IGuiLib](#).

Definition at line 50 of file RayGui.cpp.

```
50
51     return RayGUICPP::Slider::Draw({x, y, width, height}, textLeft, textRight, value, minValue,
52                                     maxValue);
```

13.78.3.15 DrawSpinner()

```
bool RayGui::DrawSpinner (
    float x,
    float y,
    float width,
    float height,
    std::shared_ptr< int > value,
    int minValue,
    int maxValue,
    bool editMode ) [override], [virtual]
```

Implements [IGuiLib](#).

Definition at line 47 of file RayGui.cpp.

```
47
48     return RayGUICPP::Spinner::Draw({x, y, width, height}, value, minValue, maxValue, editMode);
49 }
```

13.78.3.16 DrawStatusBar()

```
void RayGui::DrawStatusBar (
    float x,
    float y,
    float width,
    float height,
    const std::string & text ) [override], [virtual]
```

Implements [IGuiLib](#).

Definition at line 56 of file RayGui.cpp.

```
56
57     RayGUICPP::StatusBar::Draw({x, y, width, height}, text);
58 }
```

13.78.3.17 DrawTabBar()

```
int RayGui::DrawTabBar (
    float x,
    float y,
    float width,
    float height,
```

```

    const std::vector< std::string > & tabs,
    int active ) [override], [virtual]
Implements IGuiLib.
Definition at line 76 of file RayGui.cpp.
76
77     {
78         return RayGUICPP::TabBar::Draw({x, y, width, height}, tabs, active);
78 }

```

13.78.3.18 DrawTextBox()

```

bool RayGui::DrawTextBox (
    float x,
    float y,
    float width,
    float height,
    std::shared_ptr< std::string > text,
    int maxLength,
    bool editMode ) [override], [virtual]

```

Implements IGuiLib.

Definition at line 41 of file RayGui.cpp.

```

41
42     {
43         return RayGUICPP::TextBox::Draw({x, y, width, height}, text, maxLength, editMode);
43 }

```

13.78.3.19 DrawTextInputBox()

```

int RayGui::DrawTextInputBox (
    float x,
    float y,
    float width,
    float height,
    const std::string & title,
    const std::string & message,
    std::shared_ptr< std::string > text,
    const std::string & buttons ) [override], [virtual]

```

Implements IGuiLib.

Definition at line 88 of file RayGui.cpp.

```

88
89     {
90         return RayGUICPP::TextInputBox::Draw({x, y, width, height}, title, message, text, buttons);
90 }

```

13.78.3.20 DrawToggle()

```

bool RayGui::DrawToggle (
    float x,
    float y,
    float width,
    float height,
    const std::string & text,
    bool active ) [override], [virtual]

```

Implements IGuiLib.

Definition at line 29 of file RayGui.cpp.

```

29
30     return RayGUICPP::Toggle::Draw({x, y, width, height}, text, active);
31 }

```

13.78.3.21 DrawToggleGroup()

```
int RayGui::DrawToggleGroup (
    float x,
    float y,
    float width,
    float height,
    const std::string & text,
    int active ) [override], [virtual]
Implements IGuiLib.
Definition at line 32 of file RayGui.cpp.
32
33     {
34         return RayGUICPP::ToggleGroup::Draw({x, y, width, height}, text, active);
34 }
```

13.78.3.22 DrawValueBox()

```
bool RayGui::DrawValueBox (
    float x,
    float y,
    float width,
    float height,
    std::shared_ptr< int > value,
    int minValue,
    int maxValue,
    bool editMode ) [override], [virtual]
Implements IGuiLib.
Definition at line 44 of file RayGui.cpp.
44
45     {
46         return RayGUICPP::ValueBox::Draw({x, y, width, height}, value, minValue, maxValue, editMode);
46 }
```

13.78.3.23 DrawWindowBox()

```
bool RayGui::DrawWindowBox (
    float x,
    float y,
    float width,
    float height,
    const std::string & title ) [override], [virtual]
Implements IGuiLib.
Definition at line 59 of file RayGui.cpp.
59
60     return RayGUICPP::WindowBox::Draw({x, y, width, height}, title);
61 }
```

13.78.3.24 GetStyle()

```
int RayGui::GetStyle (
    int control,
    int property ) [override], [virtual]
Implements IGuiLib.
Definition at line 100 of file RayGui.cpp.
100
101     {
101         return RayGUICPP::RayGui::GetStyle(control, property);
102 }
```

13.78.3.25 IconText()

```
std::string RayGui::IconText (
    int iconId,
    const std::string & text ) [override], [virtual]
Implements IGuiLib.
```

Definition at line 91 of file RayGui.cpp.

```
91
92     return RayGUICPP::Icon::IconText(iconId, text);
93 }
```

13.78.3.26 LoadStyle()

```
void RayGui::LoadStyle (
    const std::string & file ) [override], [virtual]
Implements IGuiLib.
```

Definition at line 94 of file RayGui.cpp.

```
94
95     RayGUICPP::RayGui::LoadStyle(file);
96 }
```

13.78.3.27 SetStyle()

```
void RayGui::SetStyle (
    int control,
    int property,
    int value ) [override], [virtual]
```

Implements [IGuiLib](#).

Definition at line 97 of file RayGui.cpp.

```
97
98     RayGUICPP::RayGui::SetStyle(control, property, value);
99 }
```

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

- /root/Desktop/Zappy/libs/RayGUICPP/[RayGui.hpp](#)
- /root/Desktop/Zappy/libs/RayGUICPP/[RayGui.cpp](#)

13.79 RayGUICPP::RayGui Class Reference

```
#include <RayGuiEncap.hpp>
```

Public Member Functions

- [RayGui \(\)=default](#)
- [~RayGui \(\)=default](#)

Static Public Member Functions

- static void [LoadStyle](#) (const std::string &file)
- static void [SetStyle](#) (int control, int property, int value)
- static int [GetStyle](#) (int control, int property)

13.79.1 Detailed Description

Definition at line 18 of file RayGuiEncap.hpp.

13.79.2 Constructor & Destructor Documentation

13.79.2.1 RayGui()

```
RayGUICPP::RayGui::RayGui ( ) [default]
```

13.79.2.2 ~RayGui()

```
RayGUICPP::RayGui::~RayGui ( ) [default]
```

13.79.3 Member Function Documentation

13.79.3.1 GetStyle()

```
static int RayGUICPP::RayGui::GetStyle (
    int control,
    int property) [inline], [static]
Definition at line 29 of file RayGuiEncap.hpp.
29
30     return GuiGetStyle(control, property);
31 }
```

13.79.3.2 LoadStyle()

```
static void RayGUICPP::RayGui::LoadStyle (
    const std::string & file) [inline], [static]
Definition at line 23 of file RayGuiEncap.hpp.
23
24     GuiLoadStyle(file.c_str());
25 }
```

13.79.3.3 SetStyle()

```
static void RayGUICPP::RayGui::SetStyle (
    int control,
    int property,
    int value) [inline], [static]
Definition at line 26 of file RayGuiEncap.hpp.
26
27     GuiSetStyle(control, property, value);
28 }
```

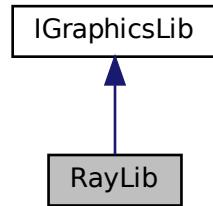
The documentation for this class was generated from the following file:

- /root/Desktop/Zappy/libs/RayGUICPP/include/RayGuiEncap.hpp

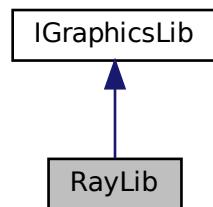
13.80 RayLib Class Reference

```
#include <RayLib.hpp>
```

Inheritance diagram for RayLib:



Collaboration diagram for RayLib:



Public Member Functions

- `RayLib ()`
- `~RayLib () override`
- `void InitWindow (int width, int height, const std::string &title) override`
- `void CloseWindow () override`
- `bool WindowShouldClose () override`
- `void setFps (int fps) override`
- `void BeginDrawing () override`
- `void EndDrawing () override`
- `void ClearBackground (ZappyTypes::Color color) override`
- `void DrawRectangle (int x, int y, int width, int height, ZappyTypes::Color color) override`
- `void DrawCircle (int centerX, int centerY, float radius, ZappyTypes::Color color) override`
- `void DrawCube (ZappyTypes::Vector3 position, float width, float height, float length, ZappyTypes::Color color) override`
- `void DrawSphere (ZappyTypes::Vector3 centerPos, float radius, ZappyTypes::Color color) override`
- `void DrawPlane (ZappyTypes::Vector3 centerPos, ZappyTypes::Vector2 size, ZappyTypes::Color color) override`
- `void DrawGrid (int slices, float spacing) override`
- `void DrawCylinder (ZappyTypes::Vector3 position, float radiusTop, float radiusBottom, float height, int slices, ZappyTypes::Color color) override`
- `void DrawLine3D (ZappyTypes::Vector3 startPos, ZappyTypes::Vector3 endPos, ZappyTypes::Color color) override`

- int `LoadTexture2D` (const std::string &path) override
- void `DrawTexture2D` (int textureId, int x, int y) override
- void `UnloadTexture2D` (int textureId) override
- bool `IsTextureReady` (int textureId) const override
- void `LoadFont` (const std::string &path) override
- void `DrawText` (const std::string &text, int x, int y, int size, `ZappyTypes::Color` color) override
- void `UnloadFont` () override
- void `DrawText3D` (const std::string &text, `ZappyTypes::Vector3` position, float fontSize, float fontSpacing, float lineSpacing, bool backface, `ZappyTypes::Color` tint) override
- void `DrawTextWave3D` (const std::string &text, `ZappyTypes::Vector3` position, float fontSize, float fontSpacing, float lineSpacing, bool backface, float time, `ZappyTypes::Color` tint) override
- `ZappyTypes::Vector3 MeasureText3D` (const std::string &text, float fontSize, float fontSpacing, float lineSpacing) override
- bool `IsKeyPressed` (int key) override
- bool `IsKeyDown` (int key) override
- bool `IsMouseButtonPressed` (int button) override
- bool `IsMouseButtonDown` (int button) override
- bool `IsMouseButtonReleased` (int button) override
- int `GetMouseX` () override
- int `GetMouseY` () override
- `ZappyTypes::Vector2 GetMousePosition` () override
- float `GetMouseWheelMove` () override
- void `PlaySound` (const std::string &path) override
- void `StopSound` () override
- void `SetSoundVolume` (float volume) override
- void `PlayMusic` (const std::string &path) override
- void `StopMusic` () override
- void `SetMusicVolume` (float volume) override
- void `UpdateMusic` () override
- void `CreateCamera2D` () override
- void `CreateCamera3D` (`ZappyTypes::Vector3` position, `ZappyTypes::Vector3` target, `ZappyTypes::Vector3` up, float fovy, int projection) override
- void `BeginCamera3D` () override
- void `EndCamera3D` () override
- void `LoadTexture3D` (const std::string &path) override
- void `BindTexture3D` (int unit=0) override
- void `UnloadTexture3D` () override
- int `LoadModel3D` (const std::string &path) override
- void `DrawModel3D` (int modelId, `ZappyTypes::Vector3` position, float scale, `ZappyTypes::Color` color) override
- void `DrawModelEx` (int modelId, `ZappyTypes::Vector3` position, `ZappyTypes::Vector3` rotationAxis, float rotationAngle, float scale) override
- void `UnloadModel3D` (int modelId) override
- int `LoadModelWithTexture` (const std::string &modelPath, const std::string &texturePath) override

13.80.1 Detailed Description

Definition at line 24 of file RayLib.hpp.

13.80.2 Constructor & Destructor Documentation

13.80.2.1 RayLib()

```
RayLib::RayLib ( )
```

Definition at line 16 of file RayLib.cpp.

```
16 { }
```

13.80.2.2 ~RayLib()

```
RayLib::~RayLib ( ) [override]
Definition at line 18 of file RayLib.cpp.
18
19     if (_initialized)
20         CloseWindow();
21 }
```

13.80.3 Member Function Documentation

13.80.3.1 BeginCamera3D()

void RayLib::BeginCamera3D () [override], [virtual]

Implements [IGraphicsLib](#).

Definition at line 255 of file RayLib.cpp.

```
255
256     if (_camera3D.has_value ()) {
257         try {
258             _camera3D->beginMode ();
259         } catch (const std::exception& e) {
260             std::cerr << "Erreur BeginCamera3D: " << e.what () << std::endl;
261         }
262     }
263 }
```

13.80.3.2 BeginDrawing()

void RayLib::BeginDrawing () [override], [virtual]

Implements [IGraphicsLib](#).

Definition at line 47 of file RayLib.cpp.

```
47
48     if (_window)
49         _window->beginDrawing ();
50 }
```

13.80.3.3 BindTexture3D()

void RayLib::BindTexture3D (

int unit = 0) [override], [virtual]

Implements [IGraphicsLib](#).

Definition at line 280 of file RayLib.cpp.

```
280
281     if (_texture3D)
282         _texture3D->bind (unit);
283 }
```

13.80.3.4 ClearBackground()

void RayLib::ClearBackground (

[ZappyTypes::Color](#) color) [override], [virtual]

Implements [IGraphicsLib](#).

Definition at line 57 of file RayLib.cpp.

```
57
58     if (_window)
59         _window->clear (TypeAdapter::ToRaylib (color));
60 }
```

13.80.3.5 CloseWindow()

void RayLib::CloseWindow () [override], [virtual]

Implements [IGraphicsLib](#).

Definition at line 28 of file RayLib.cpp.

```
28     _window.reset();
29     _initialized = false;
30     _textures.clear();
31     _models.clear();
32     _font.reset();
33     _sound.reset();
34     _music.reset();
35     _texture3D.reset();
36 }
37 }
```

13.80.3.6 CreateCamera2D()

void RayLib::CreateCamera2D () [override], [virtual]

Implements [IGraphicsLib](#).

Definition at line 238 of file RayLib.cpp.

```
238 {
239     _camera2D.emplace();
240 }
```

13.80.3.7 CreateCamera3D()

```
void RayLib::CreateCamera3D (
    ZappyTypes::Vector3 position,
    ZappyTypes::Vector3 target,
    ZappyTypes::Vector3 up,
    float fovy,
    int projection) [override], [virtual]
```

Implements [IGraphicsLib](#).

Definition at line 242 of file RayLib.cpp.

```
243 {
244     try {
245         Vector3 rayPos = TypeAdapter::ToRaylib(position);
246         Vector3 rayTarget = TypeAdapter::ToRaylib(target);
247         Vector3 rayUp = TypeAdapter::ToRaylib(up);
248         _camera3D = raylibcpp::Camera3DWrap(rayPos, rayTarget, rayUp, fovy, projection);
249     } catch (const std::exception& e) {
250         std::cerr << "Erreur création caméra 3D: " << e.what() << std::endl;
251         throw;
252     }
253 }
```

13.80.3.8 DrawCircle()

```
void RayLib::DrawCircle (
    int centerX,
    int centerY,
    float radius,
    ZappyTypes::Color color) [override], [virtual]
```

Implements [IGraphicsLib](#).

Definition at line 66 of file RayLib.cpp.

```
66 {
67     raylibcpp::Shape::drawCircle(centerX, centerY, radius, TypeAdapter::ToRaylib(color));
68 }
```

13.80.3.9 DrawCube()

```
void RayLib::DrawCube (
    ZappyTypes::Vector3 position,
    float width,
    float height,
    float length,
    ZappyTypes::Color color) [override], [virtual]
```

Implements [IGraphicsLib](#).

Definition at line 70 of file RayLib.cpp.

```
70
71     raylibcpp::Shape::drawCube(TypeAdapter::ToRaylib(position), width, height, length,
72     TypeAdapter::ToRaylib(color));
73 }
```

13.80.3.10 DrawCylinder()

```
void RayLib::DrawCylinder (
    ZappyTypes::Vector3 position,
    float radiusTop,
    float radiusBottom,
    float height,
    int slices,
    ZappyTypes::Color color ) [override], [virtual]
```

Implements [IGraphicsLib](#).

Definition at line 86 of file RayLib.cpp.

```
86
87     raylibcpp::Shape::drawCylinder(TypeAdapter::ToRaylib(position), radiusTop, radiusBottom, height,
88     slices, TypeAdapter::ToRaylib(color));
89 }
```

13.80.3.11 DrawGrid()

```
void RayLib::DrawGrid (
    int slices,
    float spacing ) [override], [virtual]
```

Implements [IGraphicsLib](#).

Definition at line 82 of file RayLib.cpp.

```
82
83     raylibcpp::Shape::drawGrid(slices, spacing);
84 }
```

13.80.3.12 DrawLine3D()

```
void RayLib::DrawLine3D (
    ZappyTypes::Vector3 startPos,
    ZappyTypes::Vector3 endPos,
    ZappyTypes::Color color ) [override], [virtual]
```

Implements [IGraphicsLib](#).

Definition at line 90 of file RayLib.cpp.

```
90
91     raylibcpp::Shape::drawLine3D(TypeAdapter::ToRaylib(startPos), TypeAdapter::ToRaylib(endPos),
92     TypeAdapter::ToRaylib(color));
93 }
```

13.80.3.13 DrawModel3D()

```
void RayLib::DrawModel3D (
    int modelId,
    ZappyTypes::Vector3 position,
    float scale,
    ZappyTypes::Color color ) [override], [virtual]
```

Implements [IGraphicsLib](#).

Definition at line 301 of file RayLib.cpp.

```
301
302     auto it = _models.find(modelId);
303     if (it != _models.end() && it->second) {
304         it->second->draw(TypeAdapter::ToRaylib(position), scale, TypeAdapter::ToRaylib(color));
305 }
```

```

305     } else {
306         std::cerr << "Tentative de dessiner un modèle inexistant (ID: " << modelId << ")" << std::endl;
307     }
308 }
```

13.80.3.14 DrawModelEx()

```
void RayLib::DrawModelEx (
    int modelId,
    ZappyTypes::Vector3 position,
    ZappyTypes::Vector3 rotationAxis,
    float rotationAngle,
    float scale ) [override], [virtual]
```

Implements [IGraphicsLib](#).

Definition at line 310 of file RayLib.cpp.

```

310
311     auto it = _models.find(modelId);
312     if (it != _models.end() && it->second) {
313         ::DrawModelEx(
314             it->second->get(),
315             TypeAdapter::ToRaylib(position),
316             TypeAdapter::ToRaylib(rotationAxis),
317             rotationAngle,
318             Vector3{scale, scale, scale},
319             WHITE
320         );
321     } else {
322         std::cerr << "Tentative de dessiner un modèle inexistant avec rotation (ID: " << modelId << ")" <<
323         std::endl;
324 }
```

13.80.3.15 DrawPlane()

```
void RayLib::DrawPlane (
    ZappyTypes::Vector3 centerPos,
    ZappyTypes::Vector2 size,
    ZappyTypes::Color color ) [override], [virtual]
```

Implements [IGraphicsLib](#).

Definition at line 78 of file RayLib.cpp.

```

78
79     raylibcpp::Shape::drawPlane(TypeAdapter::ToRaylib(centerPos), TypeAdapter::ToRaylib(size),
80     TypeAdapter::ToRaylib(color)); }
```

13.80.3.16 DrawRectangle()

```
void RayLib::DrawRectangle (
    int x,
    int y,
    int width,
    int height,
    ZappyTypes::Color color ) [override], [virtual]
```

Implements [IGraphicsLib](#).

Definition at line 62 of file RayLib.cpp.

```

62
63     raylibcpp::Shape::drawRectangle(x, y, width, height, TypeAdapter::ToRaylib(color)); }
```

13.80.3.17 DrawSphere()

```
void RayLib::DrawSphere (
    ZappyTypes::Vector3 centerPos,
```

```
    float radius,
    ZappyTypes::Color color ) [override], [virtual]
```

Implements [IGraphicsLib](#).

Definition at line 74 of file RayLib.cpp.

```
74
75     raylibcpp::Shape::drawSphere(TypeAdapter::ToRaylib(centerPos), radius, TypeAdapter::ToRaylib(color));
76 }
```

13.80.3.18 DrawText()

```
void RayLib::DrawText (
    const std::string & text,
    int x,
    int y,
    int size,
    ZappyTypes::Color color ) [override], [virtual]
```

Implements [IGraphicsLib](#).

Definition at line 130 of file RayLib.cpp.

```
130
131     if (_font.has_value())
132         ::DrawTextEx(_font.value().get(), text.c_str(), (Vector2){static_cast<float>(x),
133             static_cast<float>(y)}, static_cast<float>(size), 1.0f, TypeAdapter::ToRaylib(color));
134     else
135         ::DrawText(text.c_str(), x, y, size, TypeAdapter::ToRaylib(color));
```

13.80.3.19 DrawText3D()

```
void RayLib::DrawText3D (
    const std::string & text,
    ZappyTypes::Vector3 position,
    float fontSize,
    float fontSpacing,
    float lineSpacing,
    bool backface,
    ZappyTypes::Color tint ) [override], [virtual]
```

Implements [IGraphicsLib](#).

Definition at line 137 of file RayLib.cpp.

```
137
138     ::Font font = _font.has_value() ? _font.value().get() : GetFontDefault();
139     raylibcpp::Text3D::DrawText3D(
140         font, text, TypeAdapter::ToRaylib(position),
141         fontSize, fontSpacing, lineSpacing, backface,
142         TypeAdapter::ToRaylib(tint));
143 }
```

13.80.3.20 DrawTexture2D()

```
void RayLib::DrawTexture2D (
    int textureId,
    int x,
    int y ) [override], [virtual]
```

Implements [IGraphicsLib](#).

Definition at line 106 of file RayLib.cpp.

```
106
107     auto it = _textures.find(textureId);
108     if (it != _textures.end())
109         it->second->draw(x, y);
110 }
```

13.80.3.21 DrawTextWave3D()

```
void RayLib::DrawTextWave3D (
    const std::string & text,
    ZappyTypes::Vector3 position,
    float fontSize,
    float fontSpacing,
    float lineSpacing,
    bool backface,
    float time,
    ZappyTypes::Color tint ) [override], [virtual]
Implements IGraphicsLib.
Definition at line 145 of file RayLib.cpp.
145
146     ::Font font = _font.has_value() ? _font.value().get() : GetFontDefault();
147     auto config = std::make_shared<raylibcpp::WaveTextConfig>(raylibcpp::WaveTextConfig{
148         .waveRange = (Vector3){ 0.45f, 0.45f, 0.45f },
149         .waveSpeed = (Vector3){ 3.0f, 3.0f, 0.5f },
150         .waveOffset = (Vector3){ 0.35f, 0.35f, 0.35f }
151     });
152     raylibcpp::Text3D::DrawTextWave3D(
153         font, text, TypeAdapter::ToRaylib(position),
154         fontSize, fontSpacing, lineSpacing, backface,
155         config, time, TypeAdapter::ToRaylib(tint));
156 }
```

13.80.3.22 EndCamera3D()

```
void RayLib::EndCamera3D () [override], [virtual]
```

Implements IGraphicsLib.

Definition at line 265 of file RayLib.cpp.

```
265
266     if (_camera3D.has_value()) {
267         try {
268             _camera3D->endMode();
269         } catch (const std::exception& e) {
270             std::cerr << "Erreur EndCamera3D: " << e.what() << std::endl;
271         }
272     }
273 }
```

13.80.3.23 EndDrawing()

```
void RayLib::EndDrawing () [override], [virtual]
```

Implements IGraphicsLib.

Definition at line 52 of file RayLib.cpp.

```
52
53     if (_window)
54         _window->endDrawing();
55 }
```

13.80.3.24 GetMousePosition()

```
ZappyTypes::Vector2 RayLib::GetMousePosition () [override], [virtual]
```

Implements IGraphicsLib.

Definition at line 193 of file RayLib.cpp.

```
193
194     Vector2 pos = raylibcpp::Input::getMousePosition();
195     return TypeAdapter::FromRaylib(pos);
196 }
```

13.80.3.25 GetMouseWheelMove()

```
float RayLib::GetMouseWheelMove () [override], [virtual]
```

Implements [IGraphicsLib](#).

Definition at line 198 of file RayLib.cpp.

```
198     {
199     return raylibcpp::Input::getMouseWheelMove();
200 }
```

13.80.3.26 GetMouseX()

```
int RayLib::GetMouseX () [override], [virtual]
```

Implements [IGraphicsLib](#).

Definition at line 185 of file RayLib.cpp.

```
185     {
186     return raylibcpp::Input::getMouseX();
187 }
```

13.80.3.27 GetMouseY()

```
int RayLib::GetMouseY () [override], [virtual]
```

Implements [IGraphicsLib](#).

Definition at line 189 of file RayLib.cpp.

```
189     {
190     return raylibcpp::Input::getMouseY();
191 }
```

13.80.3.28 InitWindow()

```
void RayLib::InitWindow (
    int width,
    int height,
    const std::string & title) [override], [virtual]
```

Implements [IGraphicsLib](#).

Definition at line 23 of file RayLib.cpp.

```
23
24     _window = std::make_unique<raylibcpp::Window>(width, height, title);
25     _initialized = true;
26 }
```

13.80.3.29 IsKeyDown()

```
bool RayLib::IsKeyDown (
    int key) [override], [virtual]
```

Implements [IGraphicsLib](#).

Definition at line 169 of file RayLib.cpp.

```
169     {
170     return raylibcpp::Input::isKeyDown(key);
171 }
```

13.80.3.30 IsKeyPressed()

```
bool RayLib::IsKeyPressed (
    int key) [override], [virtual]
```

Implements [IGraphicsLib](#).

Definition at line 165 of file RayLib.cpp.

```
165     {
166     return raylibcpp::Input::isKeyPressed(key);
167 }
```

13.80.3.31 IsMouseButtonDown()

```
bool RayLib::IsMouseButtonDown (
    int button ) [override], [virtual]
Implements IGraphicsLib.
Definition at line 177 of file RayLib.cpp.
177
178     return raylibcpp::Input::isMouseButtonDown(button);
179 }
```

13.80.3.32 IsMouseButtonPressed()

```
bool RayLib::IsMouseButtonPressed (
    int button ) [override], [virtual]
Implements IGraphicsLib.
Definition at line 173 of file RayLib.cpp.
173
174     return raylibcpp::Input::isMouseButtonPressed(button);
175 }
```

13.80.3.33 IsMouseButtonReleased()

```
bool RayLib::IsMouseButtonReleased (
    int button ) [override], [virtual]
Implements IGraphicsLib.
Definition at line 181 of file RayLib.cpp.
181
182     return raylibcpp::Input::isMouseButtonReleased(button);
183 }
```

13.80.3.34 IsTextureReady()

```
bool RayLib::IsTextureReady (
    int textureId ) const [override], [virtual]
Implements IGraphicsLib.
Definition at line 116 of file RayLib.cpp.
116
117     auto it = _textures.find(textureId);
118     return it != _textures.end() && it->second->isReady();
119 }
```

13.80.3.35 LoadFont()

```
void RayLib::LoadFont (
    const std::string & path ) [override], [virtual]
Implements IGraphicsLib.
Definition at line 122 of file RayLib.cpp.
122
123     _font.emplace(path);
124 }
```

13.80.3.36 LoadModel3D()

```
int RayLib::LoadModel3D (
    const std::string & path ) [override], [virtual]
Implements IGraphicsLib.
Definition at line 290 of file RayLib.cpp.
290
291     try {
292         int modelId = _nextModelId++;
293         _models[modelId] = std::make_unique<raylibcpp::ModelWrap>(path);
```

```

294     return modelId;
295 } catch (const std::exception& e) {
296     std::cerr << "Erreur lors du chargement du modèle: " << path << " - " << e.what() << std::endl;
297     return -1;
298 }
299 }
```

13.80.3.37 LoadModelWithTexture()

```
int RayLib::LoadModelWithTexture (
    const std::string & modelPath,
    const std::string & texturePath ) [override], [virtual]
```

Implements [IGraphicsLib](#).

Definition at line 335 of file RayLib.cpp.

```

335
336     try {
337         int modelId = _nextModelId++;
338         _models[modelId] = std::make_unique<raylibcpp::ModelWrap>(modelPath);
339         if (!texturePath.empty()) {
340             Texture2D texture = ::LoadTexture(texturePath.c_str());
341             if (texture.id > 0) {
342                 if (_models[modelId]->get().materialCount > 0) {
343                     _models[modelId]->get().materials[0].maps[MATERIAL_MAP_DIFFUSE].texture = texture;
344                 } else {
345                     ::UnloadTexture(texture);
346                 }
347             }
348         }
349         return modelId;
350     } catch (const std::exception& e) {
351         std::cerr << "Erreur lors du chargement du modèle avec texture: " << modelPath << " - " << e.what()
352         << std::endl;
353         return -1;
354     }
355 }
```

13.80.3.38 LoadTexture2D()

```
int RayLib::LoadTexture2D (
    const std::string & path ) [override], [virtual]
```

Implements [IGraphicsLib](#).

Definition at line 95 of file RayLib.cpp.

```

95
96     try {
97         int id = raylibcpp::Texture::getNextId();
98         _textures[id] = std::make_unique<raylibcpp::Texture>(path);
99         return id;
100    } catch (const std::exception& e) {
101        std::cerr << "Erreur de chargement de texture: " << e.what() << std::endl;
102        return -1;
103    }
104 }
```

13.80.3.39 LoadTexture3D()

```
void RayLib::LoadTexture3D (
    const std::string & path ) [override], [virtual]
```

Implements [IGraphicsLib](#).

Definition at line 276 of file RayLib.cpp.

```

276
277     _texture3D.emplace(path);
278 }
```

13.80.3.40 MeasureText3D()

```
ZappyTypes::Vector3 RayLib::MeasureText3D (
    const std::string & text,
    float fontSize,
```

```

        float fontSpacing,
        float lineSpacing ) [override], [virtual]
Implements IGraphicsLib.
Definition at line 158 of file RayLib.cpp.
158
159     ::Font font = _font.has_value() ? _font.value().get() : GetFontDefault();
160     Vector3 size = raylibcpp::Text3D::MeasureTextWave3D(font, text, fontSize, fontSpacing, lineSpacing);
161     return TypeAdapter::FromRaylib(size);
162 }
```

13.80.3.41 PlayMusic()

```

void RayLib::PlayMusic (
    const std::string & path ) [override], [virtual]
Implements IGraphicsLib.
Definition at line 217 of file RayLib.cpp.
217
218     _music.emplace(path);
219     _music->play();
220 }
```

13.80.3.42 PlaySound()

```

void RayLib::PlaySound (
    const std::string & path ) [override], [virtual]
Implements IGraphicsLib.
Definition at line 202 of file RayLib.cpp.
202
203     _sound.emplace(path);
204     _sound->play();
205 }
```

13.80.3.43 setFps()

```

void RayLib::setFps (
    int fps ) [override], [virtual]
Implements IGraphicsLib.
Definition at line 43 of file RayLib.cpp.
43
44     _window->setFps(fps);
45 }
```

13.80.3.44 SetMusicVolume()

```

void RayLib::SetMusicVolume (
    float volume ) [override], [virtual]
Implements IGraphicsLib.
Definition at line 227 of file RayLib.cpp.
227
228     if (_music)
229         _music->setVolume(volume);
230 }
```

13.80.3.45 SetSoundVolume()

```

void RayLib::SetSoundVolume (
    float volume ) [override], [virtual]
Implements IGraphicsLib.
Definition at line 212 of file RayLib.cpp.
212
213     if (_sound)
```

```
214     _sound->setVolume(volume);  
215 }
```

13.80.3.46 StopMusic()

void RayLib::StopMusic () [override], [virtual]
Implements [IGraphicsLib](#).

Definition at line 222 of file RayLib.cpp.

```
222 {  
223     if (_music)  
224         _music->stop();  
225 }
```

13.80.3.47 StopSound()

void RayLib::StopSound () [override], [virtual]
Implements [IGraphicsLib](#).

Definition at line 207 of file RayLib.cpp.

```
207 {  
208     if (_sound)  
209         _sound->stop();  
210 }
```

13.80.3.48 UnloadFont()

void RayLib::UnloadFont () [override], [virtual]
Implements [IGraphicsLib](#).

Definition at line 126 of file RayLib.cpp.

```
126 {  
127     _font.reset();  
128 }
```

13.80.3.49 UnloadModel3D()

void RayLib::UnloadModel3D (
 int modelId) [override], [virtual]

Implements [IGraphicsLib](#).

Definition at line 326 of file RayLib.cpp.

```
326 {  
327     auto it = _models.find(modelId);  
328     if (it != _models.end()) {  
329         _models.erase(it);  
330     } else {  
331         std::cerr << "Tentative de décharger un modèle inexistant (ID: " << modelId << ")" << std::endl;  
332     }  
333 }
```

13.80.3.50 UnloadTexture2D()

void RayLib::UnloadTexture2D (
 int textureId) [override], [virtual]

Implements [IGraphicsLib](#).

Definition at line 112 of file RayLib.cpp.

```
112 {  
113     _textures.erase(textureId);  
114 }
```

13.80.3.51 UnloadTexture3D()

void RayLib::UnloadTexture3D () [override], [virtual]
Implements [IGraphicsLib](#).

Definition at line 285 of file RayLib.cpp.

```
285
286     _texture3D.reset();
287 }
```

13.80.3.52 UpdateMusic()

void RayLib::UpdateMusic () [override], [virtual]
Implements [IGraphicsLib](#).

Definition at line 232 of file RayLib.cpp.

```
232
233     if (_music)
234         _music->update();
235 }
```

13.80.3.53 WindowShouldClose()

bool RayLib::WindowShouldClose () [override], [virtual]
Implements [IGraphicsLib](#).

Definition at line 39 of file RayLib.cpp.

```
39
40     return _window ? _window->shouldClose() : true;
41 }
```

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

- /root/Desktop/Zappy/libs/RaylibCPP/[RayLib.hpp](#)
- /root/Desktop/Zappy/libs/RaylibCPP/[RayLib.cpp](#)

13.81 ZappyTypes::Rectangle Struct Reference

```
#include <Common.hpp>
```

Data Fields

- float [x](#)
- float [y](#)
- float [width](#)
- float [height](#)

13.81.1 Detailed Description

Definition at line 30 of file Common.hpp.

13.81.2 Field Documentation

13.81.2.1 [height](#)

```
float ZappyTypes::Rectangle::height
```

Definition at line 34 of file Common.hpp.

13.81.2.2 [width](#)

```
float ZappyTypes::Rectangle::width
```

Definition at line 33 of file Common.hpp.

13.81.2.3 x

```
float ZappyTypes::Rectangle::x
Definition at line 31 of file Common.hpp.
```

13.81.2.4 y

```
float ZappyTypes::Rectangle::y
Definition at line 32 of file Common.hpp.
```

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

- /root/Desktop/Zappy/src/Shared/Common.hpp

13.82 Renderer Class Reference

```
#include <Renderer.hpp>
```

Public Member Functions

- `Renderer ()`
- `~Renderer ()=default`
- `void init (std::shared_ptr< IGraphicsLib > graphics)`
- `void render (std::shared_ptr< IGraphicsLib > graphics, std::shared_ptr< IGuiLib > gui, std::shared_ptr< CameraController > camera, std::shared_ptr< UIRenderer > uiRenderer)`
- `void renderBackground (std::shared_ptr< IGraphicsLib > graphics)`
- `void renderGrid (std::shared_ptr< IGraphicsLib > graphics)`
- `void renderScene (std::shared_ptr< IGraphicsLib > graphics)`
- `void renderUI (std::shared_ptr< IGraphicsLib > graphics, std::shared_ptr< IGuiLib > gui, std::shared_ptr< CameraController > camera, std::shared_ptr< UIRenderer > uiRenderer)`
- `void renderSprite2D (int textureId, int x, int y)`
- `void renderModelFromManager (int modelId, ZappyTypes::Vector3 position, ZappyTypes::Color color={255, 255, 255, 255})`
- `int loadResourceTexture (const std::string &resourceName, const std::string &texturePath)`
- `int getResourceTextureId (const std::string &resourceName) const`
- `int getMapWidth () const`
- `int getMapHeight () const`

13.82.1 Detailed Description

Definition at line 20 of file Renderer.hpp.

13.82.2 Constructor & Destructor Documentation

13.82.2.1 Renderer()

```
Renderer::Renderer ( )
Definition at line 15 of file Renderer.cpp.
16     : m_mapWidth(20),
17     m_mapHeight(20) {
18 }
```

13.82.2.2 ~Renderer()

```
Renderer::~Renderer ( ) [default]
```

13.82.3 Member Function Documentation

13.82.3.1 getMapHeight()

```
int Renderer::getMapHeight () const [inline]
Definition at line 43 of file Renderer.hpp.
43 { return m_mapHeight; }
```

13.82.3.2 getMapWidth()

```
int Renderer::getMapWidth () const [inline]
Definition at line 42 of file Renderer.hpp.
42 { return m_mapWidth; }
```

13.82.3.3 getResourceTextureId()

```
int Renderer::getResourceTextureId (
    const std::string & resourceName ) const
Definition at line 86 of file Renderer.cpp.
86
87     auto it = m_resourceTextures.find(resourceName);
88     if (it != m_resourceTextures.end()) {
89         return it->second;
90     }
91     return -1;
92 }
```

13.82.3.4 init()

```
void Renderer::init (
    std::shared_ptr< IGraphicsLib > graphics )
Definition at line 20 of file Renderer.cpp.
20
21     graphics->InitWindow(DEFAULT_WIDTH, DEFAULT_HEIGHT, DEFAULT_TITLE);
22     m_graphicsLib = graphics;
23 }
```

13.82.3.5 loadResourceTexture()

```
int Renderer::loadResourceTexture (
    const std::string & resourceName,
    const std::string & texturePath )
Definition at line 70 of file Renderer.cpp.
70
71     auto it = m_resourceTextures.find(resourceName);
72     if (it != m_resourceTextures.end()) {
73         return it->second;
74     }
75     if (auto graphics = m_graphicsLib.lock()) {
76         int textureId = graphics->LoadTexture2D(texturePath);
77         if (textureId != -1)
78             m_resourceTextures[resourceName] = textureId;
79         else
80             std::cerr << "Échec du chargement de la texture de ressource: " << texturePath << std::endl;
81         return textureId;
82     }
83     return -1;
84 }
```

13.82.3.6 render()

```
void Renderer::render (
```

```
    std::shared_ptr< IGraphicsLib > graphics,
    std::shared_ptr< IGuiLib > gui,
    std::shared_ptr< CameraController > camera,
    std::shared_ptr< UIRenderer > uiRenderer )
```

Definition at line 25 of file `Renderer.cpp`.

```
27     {
28
29     graphics->BeginDrawing();
30     renderBackground(graphics);
31     graphics->BeginCamera3D();
32     renderGrid(graphics);
33     renderScene(graphics);
34     graphics->EndCamera3D();
35     renderUI(graphics, gui, camera, uiRenderer);
36     graphics->EndDrawing();
37 }
```

13.82.3.7 renderBackground()

```
void Renderer::renderBackground (
    std::shared_ptr< IGraphicsLib > graphics )
```

Definition at line 39 of file `Renderer.cpp`.

```
39     {
40     graphics->ClearBackground({32, 32, 64, 255});
41 }
```

13.82.3.8 renderGrid()

```
void Renderer::renderGrid (
    std::shared_ptr< IGraphicsLib > graphics )
```

Definition at line 43 of file `Renderer.cpp`.

```
43     {
44     graphics->DrawPlane({10.0f, 0.0f, 10.0f}, {20.0f, 20.0f}, {200, 200, 200, 255});
45 }
```

13.82.3.9 renderModelFromManager()

```
void Renderer::renderModelFromManager (
    int modelId,
    ZappyTypes::Vector3 position,
    ZappyTypes::Color color = {255, 255, 255, 255} )
```

Definition at line 65 of file `Renderer.cpp`.

```
65     {
66     if (auto graphics = m_graphicsLib.lock())
67         graphics->DrawModel3D(modelId, position, 1.0f, color);
68 }
```

13.82.3.10 renderScene()

```
void Renderer::renderScene (
    std::shared_ptr< IGraphicsLib > graphics )
```

Definition at line 47 of file `Renderer.cpp`.

```
47     {
48     auto& modelManager = ModelManager::getInstance();
49     (void)modelManager;
50     (void)graphics;
51     //TODO(Sam): Implement scene rendering logic
52 }
```

13.82.3.11 renderSprite2D()

```
void Renderer::renderSprite2D (
    int textureId,
```

```

        int x,
        int y )
Definition at line 59 of file Renderer.cpp.
59
60     if (auto graphics = m_graphicsLib.lock()) {
61         graphics->DrawTexture2D(textureId, x, y);
62     }
63 }
```

13.82.3.12 renderUI()

```
void Renderer::renderUI (
    std::shared_ptr< IGraphicsLib > graphics,
    std::shared_ptr< IGuiLib > gui,
    std::shared_ptr< CameraController > camera,
    std::shared_ptr< UIRenderer > uiRenderer )
```

Definition at line 54 of file Renderer.cpp.

```
55
56     uiRenderer->renderUI(graphics, gui, camera);
57 }
```

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

- /root/Desktop/Zappy/src/GUI/renderer/Renderer.hpp
- /root/Desktop/Zappy/src/GUI/renderer/Renderer.cpp

13.83 raylibcpp::RenderState Struct Reference

```
#include <Text3DHelper.hpp>
```

Data Fields

- Vector3 position
- float textOffsetX
- float textOffsetY
- int charIndex

13.83.1 Detailed Description

Definition at line 24 of file Text3DHelper.hpp.

13.83.2 Field Documentation

13.83.2.1 charIndex

```
int raylibcpp::RenderState::charIndex
Definition at line 28 of file Text3DHelper.hpp.
```

13.83.2.2 position

```
Vector3 raylibcpp::RenderState::position
Definition at line 25 of file Text3DHelper.hpp.
```

13.83.2.3 textOffsetX

```
float raylibcpp::RenderState::textOffsetX
Definition at line 26 of file Text3DHelper.hpp.
```

13.83.2.4 textOffsetY

```
float raylibcpp::RenderState::textOffsetY
```

Definition at line 27 of file Text3DHelper.hpp.

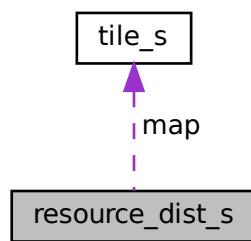
The documentation for this struct was generated from the following file:

- /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/[Text3DHelper.hpp](#)

13.84 resource_dist_s Struct Reference

```
#include <tile.h>
```

Collaboration diagram for resource_dist_s:



Data Fields

- `tile_t ** map`
- `int width`
- `int height`
- `int * tile_indices`

13.84.1 Detailed Description

Definition at line 40 of file tile.h.

13.84.2 Field Documentation

13.84.2.1 height

```
int resource_dist_s::height
```

Definition at line 43 of file tile.h.

13.84.2.2 map

```
tile_t** resource_dist_s::map
```

Definition at line 41 of file tile.h.

13.84.2.3 tile_indices

```
int* resource_dist_s::tile_indices
Definition at line 44 of file tile.h.
```

13.84.2.4 width

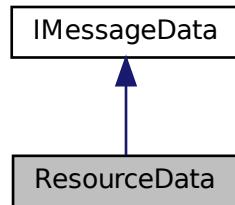
```
int resource_dist_s::width
Definition at line 42 of file tile.h.
```

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

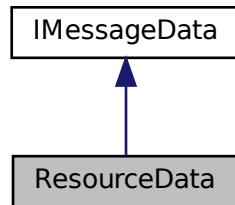
- [/root/Desktop/Zappy/src/Server/include/tile.h](#)

13.85 ResourceData Class Reference

```
#include <ResourceData.hpp>
Inheritance diagram for ResourceData:
```



Collaboration diagram for ResourceData:



Public Member Functions

- [ResourceData \(int playerId, int resourceType\)](#)
- [MessageType getType \(\) const override](#)
- [int getPlayerId \(\) const](#)
- [int getResourceType \(\) const](#)
- [void setPlayerId \(int value\)](#)
- [void setResourceType \(int value\)](#)

13.85.1 Detailed Description

Definition at line 13 of file ResourceData.hpp.

13.85.2 Constructor & Destructor Documentation

13.85.2.1 ResourceData()

```
ResourceData::ResourceData (
    int playerId,
    int resourceType ) [inline]
Definition at line 15 of file ResourceData.hpp.
16 : _playerId(playerId), _resourceType(resourceType) {}
```

13.85.3 Member Function Documentation

13.85.3.1 getPlayerId()

```
int ResourceData::getPlayerId () const [inline]
Definition at line 20 of file ResourceData.hpp.
20 { return _playerId; }
```

13.85.3.2 getResourceType()

```
int ResourceData::getResourceType () const [inline]
Definition at line 21 of file ResourceData.hpp.
21 { return _resourceType; }
```

13.85.3.3 getType()

`MessageType` ResourceData::getType () const [inline], [override], [virtual]
Implements [IMessageData](#).

Definition at line 18 of file ResourceData.hpp.

```
18 { return MessageType::Resource; }
```

13.85.3.4 setPlayerId()

```
void ResourceData::setPlayerId (
    int value ) [inline]
Definition at line 23 of file ResourceData.hpp.
23 { _playerId = value; }
```

13.85.3.5 setResourceType()

```
void ResourceData::setResourceType (
    int value ) [inline]
Definition at line 24 of file ResourceData.hpp.
24 { _resourceType = value; }
```

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

- /root/Desktop/Zappy/src/GUI/network/protocol/messageData/[ResourceData.hpp](#)

13.86 SystemWrapper::SafeBuffer Class Reference

```
#include <SystemWrapper.hpp>
```

Public Member Functions

- `SafeBuffer (size_t size)`
- `~SafeBuffer ()=default`
- `std::string & data ()`
- `const std::string & data () const`
- `size_t size () const`
- `void resize (size_t newSize)`
- `std::string toString () const`
- `void setData (const std::string &data)`

13.86.1 Detailed Description

Definition at line 73 of file SystemWrapper.hpp.

13.86.2 Constructor & Destructor Documentation

13.86.2.1 SafeBuffer()

```
SystemWrapper::SafeBuffer::SafeBuffer (
    size_t size ) [explicit]
```

Definition at line 160 of file SystemWrapper.cpp.

```
161     : _buffer(size, '\0') {}
```

13.86.2.2 ~SafeBuffer()

```
SystemWrapper::SafeBuffer::~SafeBuffer () [default]
```

13.86.3 Member Function Documentation

13.86.3.1 data() [1/2]

```
std::string & SystemWrapper::SafeBuffer::data ()
```

Definition at line 163 of file SystemWrapper.cpp.

```
163     {
164     return _buffer;
165 }
```

13.86.3.2 data() [2/2]

```
const std::string & SystemWrapper::SafeBuffer::data () const
```

Definition at line 171 of file SystemWrapper.cpp.

```
171     {
172     return _buffer;
173 }
```

13.86.3.3 resize()

```
void SystemWrapper::SafeBuffer::resize (
    size_t newSize )
Definition at line 179 of file SystemWrapper.cpp.
179
180     if (newSize == _buffer.size())
181         return;
182     _buffer.resize(newSize);
183 }
```

13.86.3.4 setData()

```
void SystemWrapper::SafeBuffer::setData (
    const std::string & data )
Definition at line 167 of file SystemWrapper.cpp.
167
168     _buffer = data;
169 }
```

13.86.3.5 size()

```
size_t SystemWrapper::SafeBuffer::size ( ) const
Definition at line 175 of file SystemWrapper.cpp.
175
176     return _buffer.size();
177 }
```

13.86.3.6 toString()

```
std::string SystemWrapper::SafeBuffer::toString ( ) const
Definition at line 185 of file SystemWrapper.cpp.
185
186     return _buffer;
187 }
```

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

- /root/Desktop/Zappy/src/GUI/network/utils/[SystemWrapper.hpp](#)
- /root/Desktop/Zappy/src/GUI/network/utils/[SystemWrapper.cpp](#)

13.87 SystemWrapper::SafePollFd Class Reference

```
#include <SystemWrapper.hpp>
```

Public Member Functions

- [SafePollFd \(\)=default](#)
- [SafePollFd \(int fd, int16_t events=POLLIN\)](#)
- pollfd & [getPollFd \(\)](#)
- const pollfd & [getPollFd \(\) const](#)
- pollfd * [getPollPtr \(\)](#)
- const pollfd * [getPollPtr \(\) const](#)
- int [getFd \(\) const](#)
- int16_t [getEvents \(\) const](#)
- int16_t [getRevents \(\) const](#)
- void [setFd \(int fd\)](#)
- void [setEvents \(int16_t events\)](#)
- void [setRevents \(int16_t revents\)](#)

13.87.1 Detailed Description

Definition at line 48 of file SystemWrapper.hpp.

13.87.2 Constructor & Destructor Documentation

13.87.2.1 SafePollFd() [1/2]

```
SystemWrapper::SafePollFd::SafePollFd ( ) [default]
```

13.87.2.2 SafePollFd() [2/2]

```
SystemWrapper::SafePollFd::SafePollFd (
    int fd,
    int16_t events = POLLIN ) [explicit]
```

Definition at line 47 of file SystemWrapper.cpp.

```
47
48     _pollfd.fd = fd;
49     _pollfd.events = events;
50     _pollfd.revents = 0;
51 }
```

13.87.3 Member Function Documentation

13.87.3.1 getEvents()

```
int16_t SystemWrapper::SafePollFd::getEvents ( ) const
Definition at line 72 of file SystemWrapper.cpp.
```

```
72
73     return _pollfd.events;
74 }
```

13.87.3.2 getFd()

```
int SystemWrapper::SafePollFd::getFd ( ) const
Definition at line 68 of file SystemWrapper.cpp.
```

```
68
69     return _pollfd.fd;
70 }
```

13.87.3.3 getPollFd() [1/2]

```
pollfd & SystemWrapper::SafePollFd::getPollFd ( )
Definition at line 53 of file SystemWrapper.cpp.
```

```
53
54     return _pollfd;
55 }
```

13.87.3.4 getPollFd() [2/2]

```
const pollfd & SystemWrapper::SafePollFd::getPollFd ( ) const
Definition at line 57 of file SystemWrapper.cpp.
```

```
57
58     return _pollfd;
59 }
```

13.87.3.5 getPollPtr() [1/2]

```
pollfd * SystemWrapper::SafePollFd::getPollPtr ( )
Definition at line 61 of file SystemWrapper.cpp.
61     {
62         return &_pollfd;
63     }
```

13.87.3.6 getPollPtr() [2/2]

```
const pollfd * SystemWrapper::SafePollFd::getPollPtr ( ) const
Definition at line 64 of file SystemWrapper.cpp.
64     {
65         return &_pollfd;
66     }
```

13.87.3.7 getRevents()

```
int16_t SystemWrapper::SafePollFd::getRevents ( ) const
Definition at line 76 of file SystemWrapper.cpp.
76     {
77         return _pollfd.revents;
78     }
```

13.87.3.8 setEvents()

```
void SystemWrapper::SafePollFd::setEvents (
    int16_t events )
Definition at line 84 of file SystemWrapper.cpp.
84     {
85         _pollfd.events = events;
86     }
```

13.87.3.9 setFd()

```
void SystemWrapper::SafePollFd::setFd (
    int fd )
Definition at line 80 of file SystemWrapper.cpp.
80     {
81         _pollfd.fd = fd;
82     }
```

13.87.3.10 setRevents()

```
void SystemWrapper::SafePollFd::setRevents (
    int16_t revents )
Definition at line 88 of file SystemWrapper.cpp.
88     {
89         _pollfd.revents = revents;
90     }
```

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

- /root/Desktop/Zappy/src/GUI/network/utils/[SystemWrapper.hpp](#)
- /root/Desktop/Zappy/src/GUI/network/utils/[SystemWrapper.cpp](#)

13.88 SystemWrapper::SafeSockAddr Class Reference

```
#include <SystemWrapper.hpp>
```

Public Member Functions

- `SafeSockAddr ()=default`
- `SafeSockAddr (const sockaddr &addr)`
- `SafeSockAddr (const sockaddr_in &addr)`
- `sockaddr_in & getAddr ()`
- `const sockaddr_in & getAddr () const`
- `sockaddr * getSockAddr ()`
- `const sockaddr * getSockAddr () const`
- `socklen_t getSize () const`

13.88.1 Detailed Description

Definition at line 29 of file SystemWrapper.hpp.

13.88.2 Constructor & Destructor Documentation

13.88.2.1 SafeSockAddr() [1/3]

```
SystemWrapper::SafeSockAddr::SafeSockAddr ( ) [default]
```

13.88.2.2 SafeSockAddr() [2/3]

```
SystemWrapper::SafeSockAddr::SafeSockAddr (
    const sockaddr & addr ) [explicit]
```

Definition at line 20 of file SystemWrapper.cpp.

```
20
21     std::memcpy(&_addr, &addr, sizeof(sockaddr));
22 }
```

13.88.2.3 SafeSockAddr() [3/3]

```
SystemWrapper::SafeSockAddr::SafeSockAddr (
    const sockaddr_in & addr ) [explicit]
```

Definition at line 24 of file SystemWrapper.cpp.

```
24
25 : _addr(addr) {
```

13.88.3 Member Function Documentation

13.88.3.1 getAddr() [1/2]

```
sockaddr_in & SystemWrapper::SafeSockAddr::getAddr ( )
```

Definition at line 27 of file SystemWrapper.cpp.

```
27
28     return _addr;
29 }
```

13.88.3.2 getAddr() [2/2]

```
const sockaddr_in & SystemWrapper::SafeSockAddr::getAddr ( ) const
```

Definition at line 31 of file SystemWrapper.cpp.

```
31
32     return _addr;
33 }
```

13.88.3.3 getSize()

```
socklen_t SystemWrapper::SafeSockAddr::getSize ( ) const
Definition at line 43 of file SystemWrapper.cpp.
43
44     return sizeof(_addr);
45 }
```

13.88.3.4 getSockAddr() [1/2]

```
sockaddr * SystemWrapper::SafeSockAddr::getSockAddr ( )
Definition at line 35 of file SystemWrapper.cpp.
35
36     return static_cast<sockaddr*>(static_cast<void*>(&_addr));
37 }
```

13.88.3.5 getSockAddr() [2/2]

```
const sockaddr * SystemWrapper::SafeSockAddr::getSockAddr ( ) const
Definition at line 39 of file SystemWrapper.cpp.
39
40     return static_cast<const sockaddr*>(static_cast<const void*>(&_addr));
41 }
```

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

- /root/Desktop/Zappy/src/GUI/network/utils/[SystemWrapper.hpp](#)
- /root/Desktop/Zappy/src/GUI/network/utils/[SystemWrapper.cpp](#)

13.89 RayGUICPP::ScrollPane Class Reference

```
#include <RayGuiEncap.hpp>
```

Static Public Member Functions

- static Rectangle [Draw](#) (const Rectangle &bounds, const std::string &text, Rectangle content, std::shared_ptr<Vector2> scroll)

13.89.1 Detailed Description

Definition at line 180 of file RayGuiEncap.hpp.

13.89.2 Member Function Documentation**13.89.2.1 Draw()**

```
static Rectangle RayGUICPP::ScrollPane::Draw (
    const Rectangle & bounds,
    const std::string & text,
    Rectangle content,
    std::shared_ptr<Vector2> scroll ) [inline], [static]
Definition at line 182 of file RayGuiEncap.hpp.
182
183     {
184         Rectangle view = {0.0f, 0.0f, 0.0f, 0.0f};
185         if (!scroll) return view;
186         Vector2 temp = *scroll;
187         GuiScrollPane(bounds, text.c_str(), content, &temp, &view);
188         *scroll = temp;
189     }
}
```

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

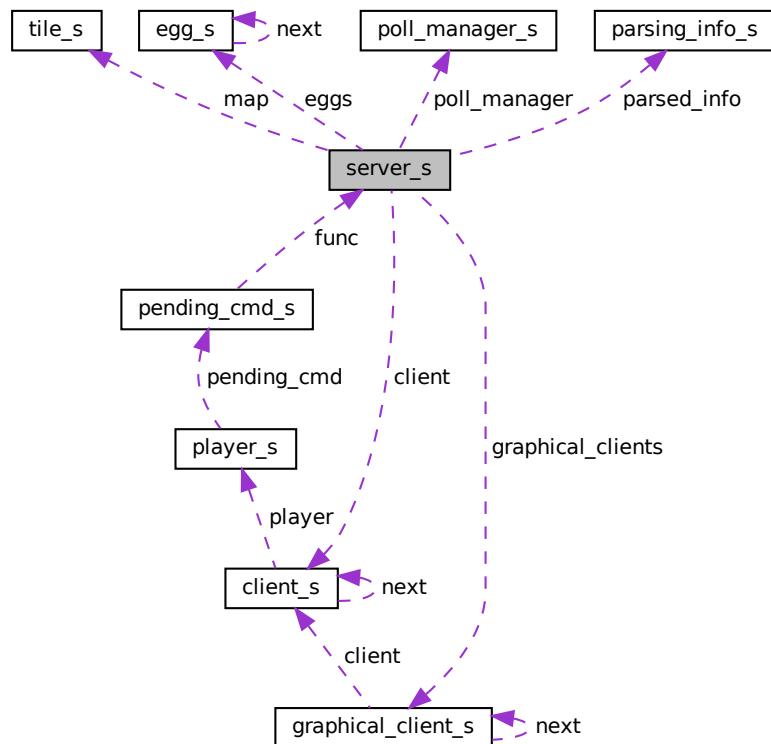
- /root/Desktop/Zappy/libs/RayGUICPP/include/RayGuiEncap.hpp

13.90 server_s Struct Reference

Structure principale du serveur Zappy.

```
#include <server.h>
```

Collaboration diagram for server_s:



Data Fields

- int `nfds`
Nombre de descripteurs de fichiers actifs.
- int `s_fd`
Socket d'écoute du serveur.
- struct `sockaddr_in` * `serv_add`
Adresse du serveur.
- `client_t` * `client`
Liste chaînée des clients IA connectés.
- `graphical_client_t` * `graphical_clients`
Liste des clients graphiques.
- `egg_t` * `eggs`
Liste des ufs présents sur la carte.
- `tile_t` ** `map`
Carte du jeu (tableau 2D de tiles)
- int `current_tick`

Tick actuel du serveur pour la synchronisation.

- `parsing_info_t * parsed_info`

Informations de configuration du serveur.

- `int * total_resources`

Nombre total de chaque type de ressource.

- `int * current_resources`

Nombre actuel de chaque type de ressource.

- `int should_run`

Flag de contrôle de la boucle principale.

- `poll_manager_t * poll_manager`

Gestionnaire de polling pour les performances.

13.90.1 Detailed Description

Structure principale du serveur [Zappy](#).

Cette structure contient tous les éléments nécessaires au fonctionnement du serveur : connexions clients, état du jeu, configuration, etc.

Definition at line 35 of file server.h.

13.90.2 Field Documentation

13.90.2.1 client

`client_t* server_s::client`

Liste chaînée des clients IA connectés.

Definition at line 39 of file server.h.

13.90.2.2 current_resources

`int* server_s::current_resources`

Nombre actuel de chaque type de ressource.

Definition at line 46 of file server.h.

13.90.2.3 current_tick

`int server_s::current_tick`

Tick actuel du serveur pour la synchronisation.

Definition at line 43 of file server.h.

13.90.2.4 eggs

`egg_t* server_s::eggs`

Liste des ufs présents sur la carte.

Definition at line 41 of file server.h.

13.90.2.5 graphical_clients

`graphical_client_t* server_s::graphical_clients`

Liste des clients graphiques.

Definition at line 40 of file server.h.

13.90.2.6 map

`tile_t** server_s::map`
Carte du jeu (tableau 2D de tiles)
Definition at line 42 of file server.h.

13.90.2.7 nfds

`int server_s::nfds`
Nombre de descripteurs de fichiers actifs.
Definition at line 36 of file server.h.

13.90.2.8 parsed_info

`parsing_info_t* server_s::parsed_info`
Informations de configuration du serveur.
Definition at line 44 of file server.h.

13.90.2.9 poll_manager

`poll_manager_t* server_s::poll_manager`
Gestionnaire de polling pour les performances.
Definition at line 48 of file server.h.

13.90.2.10 s_fd

`int server_s::s_fd`
Socket d'écoute du serveur.
Definition at line 37 of file server.h.

13.90.2.11 serv_add

`struct sockaddr_in* server_s::serv_add`
Adresse du serveur.
Definition at line 38 of file server.h.

13.90.2.12 should_run

`int server_s::should_run`
Flag de contrôle de la boucle principale.
Definition at line 47 of file server.h.

13.90.2.13 total_resources

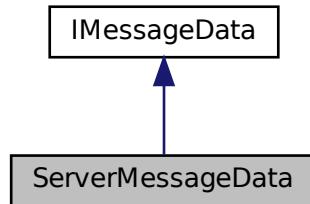
`int* server_s::total_resources`
Nombre total de chaque type de ressource.
Definition at line 45 of file server.h.
The documentation for this struct was generated from the following file:

- /root/Desktop/Zappy/src/Server/include/server.h

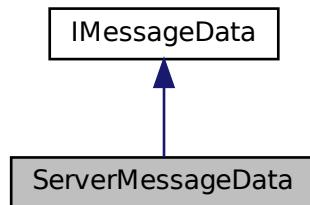
13.91 ServerMessageData Class Reference

```
#include <ServerMessageData.hpp>
```

Inheritance diagram for ServerMessageData:



Collaboration diagram for ServerMessageData:



Public Member Functions

- `ServerMessageData` (const std::string &message)
- `MessageType getType () const override`
- `const std::string & getMessage () const`
- `void setMessage (const std::string &value)`

13.91.1 Detailed Description

Definition at line 14 of file ServerMessageData.hpp.

13.91.2 Constructor & Destructor Documentation

13.91.2.1 ServerMessageData()

```
ServerMessageData::ServerMessageData (
    const std::string & message ) [inline], [explicit]
Definition at line 16 of file ServerMessageData.hpp.
17     : _message(message) {}
```

13.91.3 Member Function Documentation

13.91.3.1 getMessage()

```
const std::string& ServerMessageData::getMessage ( ) const [inline]
Definition at line 21 of file ServerMessageData.hpp.
21 { return _message; }
```

13.91.3.2 getType()

```
MessageType ServerMessageData::getType ( ) const [inline], [override], [virtual]
Implements IMESSAGEData.
Definition at line 19 of file ServerMessageData.hpp.
19 { return MessageType::ServerMessage; }
```

13.91.3.3 setMessage()

```
void ServerMessageData::setMessage (
    const std::string & value ) [inline]
Definition at line 23 of file ServerMessageData.hpp.
23 { _message = value; }
```

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

- /root/Desktop/Zappy/src/GUI/network/protocol/messageData/[ServerMessageData.hpp](#)

13.92 raylibcpp::Shape Class Reference

```
#include <Shape.hpp>
```

Static Public Member Functions

- static void [drawRectangle](#) (int x, int y, int width, int height, Color color)
- static void [drawCircle](#) (int centerX, int centerY, float radius, Color color)
- static void [drawCube](#) (Vector3 position, float width, float height, float length, Color color)
- static void [drawSphere](#) (Vector3 centerPos, float radius, Color color)
- static void [drawPlane](#) (Vector3 centerPos, Vector2 size, Color color)
- static void [drawGrid](#) (int slices, float spacing)
- static void [drawCylinder](#) (Vector3 position, float radiusTop, float radiusBottom, float height, int slices, Color color)
- static void [drawLine3D](#) (Vector3 startPos, Vector3 endPos, Color color)

13.92.1 Detailed Description

Definition at line 13 of file Shape.hpp.

13.92.2 Member Function Documentation

13.92.2.1 drawCircle()

```
void raylibcpp::Shape::drawCircle (
    int centerX,
    int centerY,
    float radius,
    Color color ) [static]
```

Definition at line 16 of file Shape.cpp.

```
16     DrawCircle(centerX, centerY, radius, color);           {  
17 }  
18 }
```

13.92.2.2 drawCube()

```
void raylibcpp::Shape::drawCube (  
    Vector3 position,  
    float width,  
    float height,  
    float length,  
    Color color ) [static]
```

Definition at line 20 of file Shape.cpp.

```
20     DrawCube(position, width, height, length, color);           {  
21 }  
22 }
```

13.92.2.3 drawCylinder()

```
void raylibcpp::Shape::drawCylinder (  
    Vector3 position,  
    float radiusTop,  
    float radiusBottom,  
    float height,  
    int slices,  
    Color color ) [static]
```

Definition at line 36 of file Shape.cpp.

```
36 {  
37     DrawCylinder(position, radiusTop, radiusBottom, height, slices, color);  
38 }
```

13.92.2.4 drawGrid()

```
void raylibcpp::Shape::drawGrid (  
    int slices,  
    float spacing ) [static]
```

Definition at line 32 of file Shape.cpp.

```
32 {  
33     DrawGrid(slices, spacing);  
34 }
```

13.92.2.5 drawLine3D()

```
void raylibcpp::Shape::drawLine3D (  
    Vector3 startPos,  
    Vector3 endPos,  
    Color color ) [static]
```

Definition at line 40 of file Shape.cpp.

```
40 {  
41     DrawLine3D(startPos, endPos, color);  
42 }
```

13.92.2.6 drawPlane()

```
void raylibcpp::Shape::drawPlane (  
    Vector3 centerPos,  
    Vector2 size,  
    Color color ) [static]
```

Definition at line 28 of file Shape.cpp.

```
28
29     DrawPlane(centerPos, size, color);
30 }
```

13.92.2.7 drawRectangle()

```
void raylibcpp::Shape::drawRectangle (
    int x,
    int y,
    int width,
    int height,
    Color color ) [static]
```

Definition at line 12 of file Shape.cpp.

```
12
13     DrawRectangle(x, y, width, height, color);
14 }
```

13.92.2.8 drawSphere()

```
void raylibcpp::Shape::drawSphere (
    Vector3 centerPos,
    float radius,
    Color color ) [static]
```

Definition at line 24 of file Shape.cpp.

```
24
25     DrawSphere(centerPos, radius, color);
26 }
```

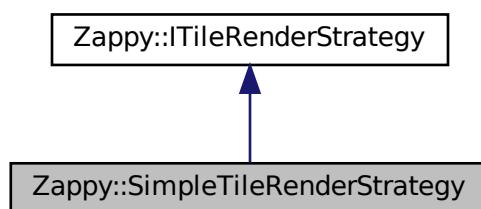
The documentation for this class was generated from the following files:

- /root/Desktop/Zappy/libs/RaylibCPP/shape/[Shape.hpp](#)
- /root/Desktop/Zappy/libs/RaylibCPP/shape/[Shape.cpp](#)

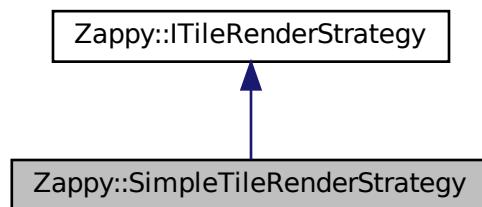
13.93 Zappy::SimpleTileRenderStrategy Class Reference

```
#include <SimpleTileRenderStrategy.hpp>
```

Inheritance diagram for Zappy::SimpleTileRenderStrategy:



Collaboration diagram for Zappy::SimpleTileRenderStrategy:



Public Member Functions

- `SimpleTileRenderStrategy (const std::shared_ptr< GraphicalContext > &ctx)`
- void `renderTile (const std::shared_ptr< IGraphicsLib > &graphicsLib, int x, int y, const ZappyTypes::Color &color, float tileSize, float spacing) override`

13.93.1 Detailed Description

Definition at line 15 of file SimpleTileRenderStrategy.hpp.

13.93.2 Constructor & Destructor Documentation

13.93.2.1 SimpleTileRenderStrategy()

```
Zappy::SimpleTileRenderStrategy::SimpleTileRenderStrategy (
    const std::shared_ptr< GraphicalContext > & ctx ) [explicit]
```

Definition at line 16 of file SimpleTileRenderStrategy.cpp.

```
17     : context(ctx) {}
```

13.93.3 Member Function Documentation

13.93.3.1 renderTile()

```
void Zappy::SimpleTileRenderStrategy::renderTile (
    const std::shared_ptr< IGraphicsLib > & graphicsLib,
    int x,
    int y,
    const ZappyTypes::Color & color,
    float tileSize,
    float spacing ) [override], [virtual]
```

Implements [Zappy::ITileRenderStrategy](#).

Definition at line 19 of file SimpleTileRenderStrategy.cpp.

```
23
24     float mapOffset = context->getMapWidth() / 2.0f;
25     ZappyTypes::Vector3 position =
26         (x - mapOffset + 0.5f) * (tileSize + spacing),
27         0.0f,
28         (y - mapOffset + 0.5f) * (tileSize + spacing)
29     };
30     graphicsLib->DrawCube(position, tileSize, tileSize * 0.1f, tileSize, color);
31     ZappyTypes::Color borderColor = {100, 100, 100, 255};
```

```

32     graphicsLib->DrawLine3D({position.x - tileSize/2, position.y, position.z - tileSize/2},
33                             {position.x + tileSize/2, position.y, position.z - tileSize/2},
34                             borderColor);
35     graphicsLib->DrawLine3D({position.x + tileSize/2, position.y, position.z - tileSize/2},
36                             {position.x + tileSize/2, position.y, position.z + tileSize/2},
37                             borderColor);
38     graphicsLib->DrawLine3D({position.x + tileSize/2, position.y, position.z + tileSize/2},
39                             {position.x - tileSize/2, position.y, position.z + tileSize/2},
40                             borderColor);
41     graphicsLib->DrawLine3D({position.x - tileSize/2, position.y, position.z + tileSize/2},
42                             {position.x - tileSize/2, position.y, position.z - tileSize/2},
43                             borderColor);
44 }

```

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

- /root/Desktop/Zappy/src/GUI/renderer/strategies/[SimpleTileRenderStrategy.hpp](#)
- /root/Desktop/Zappy/src/GUI/renderer/strategies/[SimpleTileRenderStrategy.cpp](#)

13.94 RayGUICPP::Slider Class Reference

```
#include <RayGuiEncap.hpp>
```

Static Public Member Functions

- static float [Draw](#) (const Rectangle &bounds, const std::string &textLeft, const std::string &textRight, float value, float minValue, float maxValue)

13.94.1 Detailed Description

Definition at line 134 of file RayGuiEncap.hpp.

13.94.2 Member Function Documentation

13.94.2.1 Draw()

```

static float RayGUICPP::Slider::Draw (
    const Rectangle & bounds,
    const std::string & textLeft,
    const std::string & textRight,
    float value,
    float minValue,
    float maxValue ) [inline], [static]

```

Definition at line 136 of file RayGuiEncap.hpp.

```

136
137     float v = value;
138     GuiSlider(bounds, textLeft.c_str(), textRight.c_str(), &v, minValue, maxValue);
139     return v;
140 }
```

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

- /root/Desktop/Zappy/libs/RayGUICPP/include/[RayGuiEncap.hpp](#)

13.95 socketManager.SocketManager Class Reference

Public Member Functions

- def [__init__](#) (self, sock)
- def [start](#) (self)
- def [stop](#) (self)
- def [send_command](#) (self, command, timeout=2.0)
- def [get_message](#) (self, timeout=None)
- def [has_messages](#) (self)

Data Fields

- [sock](#)
- [message_queue](#)
- [pending_requests](#)
- [running](#)
- [thread](#)
- [buffer](#)
- [lock](#)

13.95.1 Detailed Description

Definition at line 6 of file `socketManager.py`.

13.95.2 Constructor & Destructor Documentation

13.95.2.1 `__init__()`

```
def socketManager.SocketManager.__init__ (
    self,
    sock )
```

Definition at line 7 of file `socketManager.py`.

```
7  def __init__(self, sock):
8      self.sock = sock
9      self.message_queue = queue.Queue()
10     self.pending_requests = {}
11     self.running = False
12     self.thread = None
13     self.buffer = ""
14     self.lock = threading.Lock()
```

13.95.3 Member Function Documentation

13.95.3.1 `get_message()`

```
def socketManager.SocketManager.get_message (
    self,
    timeout = None )
```

Definition at line 108 of file `socketManager.py`.

```
108  def get_message(self, timeout=None):
109      try:
110          return self.message_queue.get(timeout=timeout)
111      except queue.Empty:
112          return None
113
114
```

13.95.3.2 `has_messages()`

```
def socketManager.SocketManager.has_messages (
    self )
```

Definition at line 115 of file `socketManager.py`.

```
115  def has_messages(self):
116      return not self.message_queue.empty()
```

13.95.3.3 send_command()

```
def socketManager.SocketManager.send_command (
    self,
    command,
    timeout = 2.0 )
Definition at line 79 of file socketManager.py.
79  def send_command(self, command, timeout=2.0):
80      if command is None:
81          print("Command is None, not sending.")
82          return None
83
84      request_id = str(uuid.uuid4())
85      response_queue = queue.Queue()
86
87      with self.lock:
88          self.pending_requests[request_id] = response_queue
89
90      try:
91          self.sock.send(f"{command}\n".encode('utf-8'))
92      try:
93          response = response_queue.get(timeout=timeout)
94          return response
95      except queue.Empty:
96          with self.lock:
97              self.pending_requests.pop(request_id, None)
98              print(f"Response to command '{command}' timed out.")
99          return None
100     except socket.error as e:
101         with self.lock:
102             self.pending_requests.pop(request_id, None)
103             print(f"Error sending command '{command}': {e}")
104         self.running = False
105     return None
106
107
```

13.95.3.4 start()

```
def socketManager.SocketManager.start (
    self )
Definition at line 17 of file socketManager.py.
17  def start(self):
18      self.running = True
19      self.thread = threading.Thread(target=self._listen_loop, daemon=True)
20      self.thread.start()
21
22
```

13.95.3.5 stop()

```
def socketManager.SocketManager.stop (
    self )
Definition at line 23 of file socketManager.py.
23  def stop(self):
24      self.running = False
25      if self.thread:
26          self.thread.join()
27
28
29 # fonction bloquante
```

13.95.4 Field Documentation

13.95.4.1 buffer

socketManager.SocketManager.buffer
 Definition at line 13 of file socketManager.py.

13.95.4.2 lock

socketManager.SocketManager.lock

Definition at line 14 of file socketManager.py.

13.95.4.3 message_queue

socketManager.SocketManager.message_queue

Definition at line 9 of file socketManager.py.

13.95.4.4 pending_requests

socketManager.SocketManager.pending_requests

Definition at line 10 of file socketManager.py.

13.95.4.5 running

socketManager.SocketManager.running

Definition at line 11 of file socketManager.py.

13.95.4.6 sock

socketManager.SocketManager.sock

Definition at line 8 of file socketManager.py.

13.95.4.7 thread

socketManager.SocketManager.thread

Definition at line 12 of file socketManager.py.

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

- /root/Desktop/Zappy/src/AI/agent/socketManager.py

13.96 raylibcpp::SoundWrap Class Reference

#include <Audio.hpp>

Public Member Functions

- [SoundWrap](#) (const std::string &path)
- [~SoundWrap](#) ()
- void [play](#) () const
- void [stop](#) () const
- void [pause](#) () const
- void [resume](#) () const
- void [setVolume](#) (float volume) const
- Sound & [get](#) ()

13.96.1 Detailed Description

Definition at line 20 of file Audio.hpp.

13.96.2 Constructor & Destructor Documentation

13.96.2.1 SoundWrap()

```
raylibcpp::SoundWrap::SoundWrap (
    const std::string & path ) [explicit]
Definition at line 22 of file Audio.cpp.
22
23     sound = LoadSound(path.c_str());
24     if (sound.frameCount == 0)
25         throw std::runtime_error("Erreur chargement sound: " + path);
26 }
```

13.96.2.2 ~SoundWrap()

```
raylibcpp::SoundWrap::~SoundWrap ( )
Definition at line 28 of file Audio.cpp.
28
29     UnloadSound(sound);
30 }
```

13.96.3 Member Function Documentation

13.96.3.1 get()

```
Sound & raylibcpp::SoundWrap::get ( )
Definition at line 52 of file Audio.cpp.
52
53     {
54     return sound;
```

13.96.3.2 pause()

```
void raylibcpp::SoundWrap::pause ( ) const
Definition at line 40 of file Audio.cpp.
40
41     PauseSound(sound);
42 }
```

13.96.3.3 play()

```
void raylibcpp::SoundWrap::play ( ) const
Definition at line 32 of file Audio.cpp.
32
33     PlaySound(sound);
34 }
```

13.96.3.4 resume()

```
void raylibcpp::SoundWrap::resume ( ) const
Definition at line 44 of file Audio.cpp.
44
45     ResumeSound(sound);
46 }
```

13.96.3.5 setVolume()

```
void raylibcpp::SoundWrap::setVolume (
    float volume ) const
Definition at line 48 of file Audio.cpp.
48
49     SetSoundVolume(sound, volume);
50 }
```

13.96.3.6 stop()

```
void raylibcpp::SoundWrap::stop ( ) const
Definition at line 36 of file Audio.cpp.
36
37     StopSound(sound);
38 }
```

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

- /root/Desktop/Zappy/libs/RaylibCPP/audio/[Audio.hpp](#)
- /root/Desktop/Zappy/libs/RaylibCPP/audio/[Audio.cpp](#)

13.97 RayGUICPP::Spinner Class Reference

```
#include <RayGuiEncap.hpp>
```

Static Public Member Functions

- static bool [Draw](#) (const Rectangle &bounds, std::shared_ptr< int > value, int minValue, int maxValue, bool editMode)

13.97.1 Detailed Description

Definition at line 123 of file RayGuiEncap.hpp.

13.97.2 Member Function Documentation**13.97.2.1 Draw()**

```
static bool RayGUICPP::Spinner::Draw (
    const Rectangle & bounds,
    std::shared_ptr< int > value,
    int minValue,
    int maxValue,
    bool editMode ) [inline], [static]
Definition at line 125 of file RayGuiEncap.hpp.
125
126     {
127         if (!value) return false;
128         int temp = *value;
129         bool changed = GuiSpinner(bounds, nullptr, &temp, minValue, maxValue, editMode);
130         if (changed) *value = temp;
131     }

```

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

- /root/Desktop/Zappy/libs/RayGUICPP/include/[RayGuiEncap.hpp](#)

13.98 RayGUICPP::StatusBar Class Reference

```
#include <RayGuiEncap.hpp>
```

Static Public Member Functions

- static void [Draw](#) (const Rectangle &bounds, const std::string &text)

13.98.1 Detailed Description

Definition at line 152 of file RayGuiEncap.hpp.

13.98.2 Member Function Documentation

13.98.2.1 Draw()

```
static void RayGUICPP::StatusBar::Draw (
    const Rectangle & bounds,
    const std::string & text) [inline], [static]
```

Definition at line 154 of file RayGuiEncap.hpp.

```
154
155     GuiStatusBar(bounds, text.c_str());
156 }
```

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

- /root/Desktop/Zappy/libs/RayGUICPP/include/[RayGuiEncap.hpp](#)

13.99 RayGUICPP::TabBar Class Reference

```
#include <RayGuiEncap.hpp>
```

Static Public Member Functions

- static int [Draw](#) (const Rectangle &bounds, const std::vector< std::string > &tabs, int active)

13.99.1 Detailed Description

Definition at line 192 of file RayGuiEncap.hpp.

13.99.2 Member Function Documentation

13.99.2.1 Draw()

```
static int RayGUICPP::TabBar::Draw (
    const Rectangle & bounds,
    const std::vector< std::string > & tabs,
    int active) [inline], [static]
```

Definition at line 194 of file RayGuiEncap.hpp.

```
194
195     std::vector<const char*> c_tabs;
196     for (const auto& t : tabs) c_tabs.push_back(t.c_str());
197     int value = active;
198     GuiTabBar(bounds, c_tabs.data(), static_cast<int>(c_tabs.size()), &value);
199     return value;
200 }
```

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

- /root/Desktop/Zappy/libs/RayGUICPP/include/[RayGuiEncap.hpp](#)

13.100 TcpConnection Class Reference

```
#include <TcpConnection.hpp>
```

Public Member Functions

- `TcpConnection` (`size_t initialBufferSize=4096`)
- `~TcpConnection ()`
- `void connect (const std::string &host, int port)`
- `void send (const std::string &message)`
- `std::string receive ()`
- `void close ()`
- `bool isConnected () const`
- `bool hasData () const`

13.100.1 Detailed Description

Definition at line 16 of file `TcpConnection.hpp`.

13.100.2 Constructor & Destructor Documentation

13.100.2.1 `TcpConnection()`

```
TcpConnection::TcpConnection (
    size_t initialBufferSize = 4096 ) [explicit]
```

Definition at line 15 of file `TcpConnection.cpp`.

```
15
16     _pollfd = std::make_unique<SystemWrapper::SafePollFd>(-1);
17     _sockaddr = std::make_unique<SystemWrapper::SafeSockAddr>();
18     _recvBuffer = std::make_unique<SystemWrapper::SafeBuffer>(_initialBufferSize);
19 }
```

13.100.2.2 `~TcpConnection()`

```
TcpConnection::~TcpConnection ( )
```

Definition at line 21 of file `TcpConnection.cpp`.

```
21
22     close();
23 }
```

13.100.3 Member Function Documentation

13.100.3.1 `close()`

```
void TcpConnection::close ( )
```

Definition at line 172 of file `TcpConnection.cpp`.

```
172
173     if (_socket >= 0) {
174         SystemWrapper::closeSocket (_socket);
175         _socket = -1;
176         _pollfd = std::make_unique<SystemWrapper::SafePollFd>(-1);
177     }
178 }
```

13.100.3.2 `connect()`

```
void TcpConnection::connect (
    const std::string & host,
    int port )
```

Definition at line 25 of file `TcpConnection.cpp`.

```
25
26     if (_socket >= 0)
{
```

```

27     close();
28     createAndConfigureSocket();
29     resolveAddress(host, port);
30     performConnect();
31     waitForConnection(5000);
32     _pollfd = std::make_unique<SystemWrapper::SafePollFd>(_socket, POLLIN | POLLOUT);
33 }

```

13.100.3.3 hasData()

```

bool TcpConnection::hasData () const
Definition at line 184 of file TcpConnection.cpp.
184 {
185     return true;
186 }

```

13.100.3.4 isConnected()

```

bool TcpConnection::isConnected () const
Definition at line 180 of file TcpConnection.cpp.
180 {
181     return _socket >= 0;
182 }

```

13.100.3.5 receive()

```

std::string TcpConnection::receive ()
Definition at line 117 of file TcpConnection.cpp.
117 {
118     if (_socket < 0)
119         throw TcpConnectionException("Socket is not connected");
120     try {
121         if (!waitForReadData(500))
122             return "";
123         return readDataFromSocket();
124     } catch (const std::exception& e) {
125         throw TcpConnectionException(std::string("Receive failed: ") + e.what());
126     }
127 }

```

13.100.3.6 send()

```

void TcpConnection::send (
    const std::string & message )
Definition at line 89 of file TcpConnection.cpp.
89 {
90     if (_socket < 0)
91         throw TcpConnectionException("Socket is not connected");
92     std::string finalMessage = message;
93     if (finalMessage.empty() || finalMessage.back() != '\n') {
94         finalMessage += '\n';
95     }
96     size_t totalSent = 0;
97     const size_t size = finalMessage.size();
98     while (totalSent < size) {
99         _pollfd->setEvents(POLLOUT);
100     int pollResult = SystemWrapper::pollSocket(*_pollfd, 5000);
101     if (pollResult <= 0)
102         throw TcpConnectionException("Send timeout or poll error");
103     if (!(_pollfd->getRevents() & POLLOUT))
104         throw TcpConnectionException("Socket not ready for writing");
105     SystemWrapper::SafeBuffer sendBuffer(size - totalSent);
106     sendBuffer.setData(std::string(finalMessage.c_str()) + totalSent, size - totalSent);
107     ssize_t sent = SystemWrapper::writeSocket(_socket, sendBuffer, size - totalSent);
108     if (sent < 0) {
109         if (errno == EINTR || errno == EAGAIN || errno == EWOULDBLOCK)
110             continue;
111         throw TcpConnectionException("Send error: " + SystemWrapper::getErrorString());
112     }
113     totalSent += sent;
114 }
115 }

```

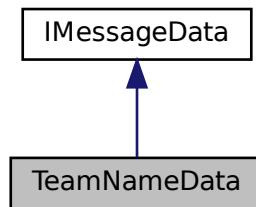
The documentation for this class was generated from the following files:

- /root/Desktop/Zappy/src/GUI/network/connection/TcpConnection.hpp
- /root/Desktop/Zappy/src/GUI/network/connection/TcpConnection.cpp

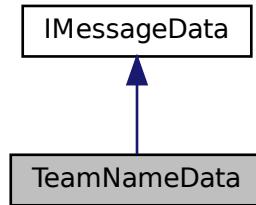
13.101 TeamNameData Class Reference

```
#include <TeamNameData.hpp>
```

Inheritance diagram for TeamNameData:



Collaboration diagram for TeamNameData:



Public Member Functions

- `TeamNameData (const std::vector< std::string > &names)`
- `MessageType getType () const override`
- `const std::vector< std::string > & getNames () const`
- `void setNames (const std::vector< std::string > &value)`
- `void addName (const std::string &name)`

13.101.1 Detailed Description

Definition at line 15 of file TeamNameData.hpp.

13.101.2 Constructor & Destructor Documentation

13.101.2.1 TeamNameData()

```
TeamNameData::TeamNameData (
    const std::vector< std::string > & names ) [inline], [explicit]
Definition at line 17 of file TeamNameData.hpp.
18     : _names(names) {}
```

13.101.3 Member Function Documentation

13.101.3.1 addName()

```
void TeamNameData::addName (
    const std::string & name ) [inline]
Definition at line 25 of file TeamNameData.hpp.
25 { _names.push_back(name); }
```

13.101.3.2 getNames()

```
const std::vector<std::string>& TeamNameData::getNames () const [inline]
Definition at line 22 of file TeamNameData.hpp.
22 { return _names; }
```

13.101.3.3 getType()

```
MessageType TeamNameData::getType () const [inline], [override], [virtual]
Implements IMessageData.
Definition at line 20 of file TeamNameData.hpp.
20 { return MessageType::TeamName; }
```

13.101.3.4 setNames()

```
void TeamNameData::setNames (
    const std::vector< std::string > & value ) [inline]
Definition at line 24 of file TeamNameData.hpp.
24 { _names = value; }
```

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

- /root/Desktop/Zappy/src/GUI/network/protocol/messageData/[TeamNameData.hpp](#)

13.102 raylibcpp::Text3D Class Reference

```
#include <Text3D.hpp>
```

Public Member Functions

- [Text3D \(\)](#)
- [~Text3D \(\)](#)

Static Public Member Functions

- static void [DrawTextCodepoint3D](#) ([::Font](#) font, int codepoint, [Vector3](#) position, float fontSize, bool backface, [Color](#) tint)
- static void [DrawText3D](#) ([::Font](#) font, const std::string &text, [Vector3](#) position, float fontSize, float fontSpacing, float lineSpacing, bool backface, [Color](#) tint)
- static void [DrawTextWave3D](#) ([::Font](#) font, const std::string &text, [Vector3](#) position, float fontSize, float fontSpacing, float lineSpacing, bool backface, const [WaveTextConfigPtr](#) &config, float time, [Color](#) tint)

- static Vector3 [MeasureTextWave3D](#) (::Font font, const std::string &text, float fontSize, float fontSpacing, float lineSpacing)

13.102.1 Detailed Description

Definition at line 17 of file Text3D.hpp.

13.102.2 Constructor & Destructor Documentation

13.102.2.1 Text3D()

```
raylibcpp::Text3D::Text3D ( )  
Definition at line 20 of file Text3D.cpp.  
20 { }
```

13.102.2.2 ~Text3D()

```
raylibcpp::Text3D::~Text3D ( )  
Definition at line 22 of file Text3D.cpp.  
22 { }
```

13.102.3 Member Function Documentation

13.102.3.1 DrawText3D()

```
void raylibcpp::Text3D::DrawText3D (  
    ::Font font,  
    const std::string & text,  
    Vector3 position,  
    float fontSize,  
    float fontSpacing,  
    float lineSpacing,  
    bool backface,  
    Color tint ) [static]  
Definition at line 29 of file Text3D.cpp.  
31 {  
32     std::cout << "DrawText3D: " << text << std::endl;  
33     Text3DRenderer::renderBasicText(font, text, position, fontSize,  
34                                         fontSpacing, lineSpacing, backface, tint);  
35 }
```

13.102.3.2 DrawTextCodepoint3D()

```
void raylibcpp::Text3D::DrawTextCodepoint3D (  
    ::Font font,  
    int codepoint,  
    Vector3 position,  
    float fontSize,  
    bool backface,  
    Color tint ) [static]  
Definition at line 24 of file Text3D.cpp.  
25 {  
26     Text3DCodepoint::draw(font, codepoint, position, fontSize, backface, tint);  
27 }
```

13.102.3.3 DrawTextWave3D()

```
void raylibcpp::Text3D::DrawTextWave3D (
    ::Font font,
    const std::string & text,
    Vector3 position,
    float fontSize,
    float fontSpacing,
    float lineSpacing,
    bool backface,
    const WaveTextConfigPtr & config,
    float time,
    Color tint ) [static]
```

Definition at line 37 of file Text3D.cpp.

```
39
40     std::cout << "DrawTextWave3D (smart pointer): " << text << std::endl;
41     Text3DParser::renderWaveText(font, text, position, fontSize, fontSpacing,
42                                   lineSpacing, backface, config, time, tint);
43 }
```

13.102.3.4 MeasureTextWave3D()

```
Vector3 raylibcpp::Text3D::MeasureTextWave3D (
    ::Font font,
    const std::string & text,
    float fontSize,
    float fontSpacing,
    float lineSpacing ) [static]
```

Definition at line 45 of file Text3D.cpp.

```
46
47     std::cout << "MeasureTextWave3D: " << text << std::endl;
48     return Text3DMeasurement::measureText(font, text, fontSize, fontSpacing, lineSpacing);
49 }
```

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

- /root/Desktop/Zappy/libs/RaylibCPP/font/Text3D.hpp
- /root/Desktop/Zappy/libs/RaylibCPP/font/Text3D.cpp

13.103 raylibcpp::Text3DCodepoint Class Reference

```
#include <Text3DCodepoint.hpp>
```

Static Public Member Functions

- static void [draw](#) (const ::Font &font, int codepoint, Vector3 position, float fontSize, bool backface, Color tint)

13.103.1 Detailed Description

Definition at line 14 of file Text3DCodepoint.hpp.

13.103.2 Member Function Documentation

13.103.2.1 draw()

```
void raylibcpp::Text3DCodepoint::draw (
    const ::Font & font,
    int codepoint,
    Vector3 position,
    float fontSize,
```

```

        bool backface,
        Color tint ) [static]
Definition at line 15 of file Text3DCodepoint.cpp.
16
17     if (font.texture.id <= 0)
18         return;
19     int index = Text3DHelper::getGlyphIndex(font, codepoint);
20     float scale = Text3DHelper::calculateScale(font, fontSize);
21     Vector3 glyphPosition = Text3DHelper::calculateGlyphPosition(font, index, position, scale);
22     Rectangle srcRec = Text3DHelper::calculateSourceRectangle(font, index);
23
24     auto dimensions = Text3DHelper::calculateGlyphDimensions(font, index, scale);
25     auto texCoords = Text3DHelper::calculateTextureCoordinates(font, srcRec);
26
27     drawDebugBoundary(glyphPosition, dimensions.width, dimensions.height);
28     renderQuad(font, glyphPosition, dimensions.width, dimensions.height,
29                 texCoords.tx, texCoords.ty, texCoords.tw, texCoords.th, backface, tint);
30 }

```

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

- /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DCodepoint.hpp
- /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DCodepoint.cpp

13.104 raylibcpp::Text3DConfig Class Reference

```
#include <Text3DConfig.hpp>
```

Static Public Member Functions

- static WaveTextConfigPtr createDefaultWaveConfig ()
- static WaveTextConfigPtr createWaveConfig (Vector3 range, Vector3 speed, Vector3 offset)
- static bool isValidWaveConfig (const WaveTextConfigPtr &config)
- static WaveTextConfigPtr createSubtleWave ()
- static WaveTextConfigPtr createIntenseWave ()
- static WaveTextConfigPtr createSlowWave ()
- static WaveTextConfigPtr createFastWave ()
- static void adjustWaveIntensity (const WaveTextConfigPtr &config, float multiplier)
- static void adjustWaveSpeed (const WaveTextConfigPtr &config, float multiplier)
- static void setShowLetterBoundary (bool show)
- static void setShowTextBoundary (bool show)
- static bool isShowLetterBoundary ()
- static bool isShowTextBoundary ()

13.104.1 Detailed Description

Definition at line 15 of file Text3DConfig.hpp.

13.104.2 Member Function Documentation

13.104.2.1 adjustWaveIntensity()

```
void raylibcpp::Text3DConfig::adjustWaveIntensity (
    const WaveTextConfigPtr & config,
    float multiplier ) [static]
```

Definition at line 82 of file Text3DConfig.cpp.

```

82
83     if (!config) return;
84     config->waveRange.x = std::clamp(config->waveRange.x * multiplier, MIN_WAVE_RANGE, MAX_WAVE_RANGE);
85     config->waveRange.y = std::clamp(config->waveRange.y * multiplier, MIN_WAVE_RANGE, MAX_WAVE_RANGE);
86     config->waveRange.z = std::clamp(config->waveRange.z * multiplier, MIN_WAVE_RANGE, MAX_WAVE_RANGE);
87 }
```

13.104.2.2 adjustWaveSpeed()

```
void raylibcpp::Text3DConfig::adjustWaveSpeed (
    const WaveTextConfigPtr & config,
    float multiplier ) [static]
Definition at line 89 of file Text3DConfig.cpp.
89
90     if (!config) return;
91     config->waveSpeed.x = std::clamp(config->waveSpeed.x * multiplier, MIN_WAVE_SPEED, MAX_WAVE_SPEED);
92     config->waveSpeed.y = std::clamp(config->waveSpeed.y * multiplier, MIN_WAVE_SPEED, MAX_WAVE_SPEED);
93     config->waveSpeed.z = std::clamp(config->waveSpeed.z * multiplier, MIN_WAVE_SPEED, MAX_WAVE_SPEED);
94 }
```

13.104.2.3 createDefaultWaveConfig()

```
WaveTextConfigPtr raylibcpp::Text3DConfig::createDefaultWaveConfig () [static]
```

Definition at line 14 of file Text3DConfig.cpp.

```
14
15     return std::make_shared<WaveTextConfig>(WaveTextConfig{
16         .waveRange = {0.45f, 0.45f, 0.45f},
17         .waveSpeed = {3.0f, 3.0f, 0.5f},
18         .waveOffset = {0.35f, 0.35f, 0.35f}
19     });
20 }
```

13.104.2.4 createFastWave()

```
WaveTextConfigPtr raylibcpp::Text3DConfig::createFastWave () [static]
```

Definition at line 74 of file Text3DConfig.cpp.

```
74
75     return std::make_shared<WaveTextConfig>(WaveTextConfig{
76         .waveRange = {0.3f, 0.3f, 0.2f},
77         .waveSpeed = {8.0f, 8.0f, 1.5f},
78         .waveOffset = {0.1f, 0.1f, 0.1f}
79     });
80 }
```

13.104.2.5 createIntenseWave()

```
WaveTextConfigPtr raylibcpp::Text3DConfig::createIntenseWave () [static]
```

Definition at line 58 of file Text3DConfig.cpp.

```
58
59     return std::make_shared<WaveTextConfig>(WaveTextConfig{
60         .waveRange = {1.0f, 1.0f, 0.8f},
61         .waveSpeed = {5.0f, 5.0f, 1.0f},
62         .waveOffset = {0.2f, 0.2f, 0.2f}
63     });
64 }
```

13.104.2.6 createSlowWave()

```
WaveTextConfigPtr raylibcpp::Text3DConfig::createSlowWave () [static]
```

Definition at line 66 of file Text3DConfig.cpp.

```
66
67     return std::make_shared<WaveTextConfig>(WaveTextConfig{
68         .waveRange = {0.6f, 0.6f, 0.4f},
69         .waveSpeed = {1.0f, 1.0f, 0.2f},
70         .waveOffset = {0.8f, 0.8f, 0.8f}
71     });
72 }
```

13.104.2.7 createSubtleWave()

```
WaveTextConfigPtr raylibcpp::Text3DConfig::createSubtleWave () [static]
```

Definition at line 50 of file Text3DConfig.cpp.

```
50
51     return std::make_shared<WaveTextConfig>(WaveTextConfig{
52         .waveRange = {0.1f, 0.1f, 0.1f},
53         .waveSpeed = {0.1f, 0.1f, 0.1f},
54         .waveOffset = {0.1f, 0.1f, 0.1f}
55     });
56 }
```

```

51     return std::make_shared<WaveTextConfig>(WaveTextConfig{
52         .waveRange = {0.2f, 0.2f, 0.1f},
53         .waveSpeed = {1.5f, 1.5f, 0.3f},
54         .waveOffset = {0.5f, 0.5f, 0.5f}
55     });
56 }

```

13.104.2.8 createWaveConfig()

```
WaveTextConfigPtr raylibcpp::Text3DConfig::createWaveConfig (
    Vector3 range,
    Vector3 speed,
    Vector3 offset ) [static]
```

Definition at line 22 of file Text3DConfig.cpp.

```

22
23     range.x = std::clamp(range.x, MIN_WAVE_RANGE, MAX_WAVE_RANGE);
24     range.y = std::clamp(range.y, MIN_WAVE_RANGE, MAX_WAVE_RANGE);
25     range.z = std::clamp(range.z, MIN_WAVE_RANGE, MAX_WAVE_RANGE);
26     speed.x = std::clamp(speed.x, MIN_WAVE_SPEED, MAX_WAVE_SPEED);
27     speed.y = std::clamp(speed.y, MIN_WAVE_SPEED, MAX_WAVE_SPEED);
28     speed.z = std::clamp(speed.z, MIN_WAVE_SPEED, MAX_WAVE_SPEED);
29     offset.x = std::clamp(offset.x, MIN_WAVE_OFFSET, MAX_WAVE_OFFSET);
30     offset.y = std::clamp(offset.y, MIN_WAVE_OFFSET, MAX_WAVE_OFFSET);
31     offset.z = std::clamp(offset.z, MIN_WAVE_OFFSET, MAX_WAVE_OFFSET);
32     return std::make_shared<WaveTextConfig>(WaveTextConfig(range, speed, offset));
33 }
```

13.104.2.9 isShowLetterBoundary()

```
bool raylibcpp::Text3DConfig::isShowLetterBoundary () [static]
```

Definition at line 107 of file Text3DConfig.cpp.

```

107
108     return _showLetterBoundary;
109 }
```

13.104.2.10 isShowTextBoundary()

```
bool raylibcpp::Text3DConfig::isShowTextBoundary () [static]
```

Definition at line 111 of file Text3DConfig.cpp.

```

111
112     return _showTextBoundary;
113 }
```

13.104.2.11 isValidWaveConfig()

```
bool raylibcpp::Text3DConfig::isValidWaveConfig (
    const WaveTextConfigPtr & config ) [static]
```

Definition at line 35 of file Text3DConfig.cpp.

```

35
36     if (!config)
37         return false;
38     if (config->waveRange.x < MIN_WAVE_RANGE || config->waveRange.x > MAX_WAVE_RANGE) return false;
39     if (config->waveRange.y < MIN_WAVE_RANGE || config->waveRange.y > MAX_WAVE_RANGE) return false;
40     if (config->waveRange.z < MIN_WAVE_RANGE || config->waveRange.z > MAX_WAVE_RANGE) return false;
41     if (config->waveSpeed.x < MIN_WAVE_SPEED || config->waveSpeed.x > MAX_WAVE_SPEED) return false;
42     if (config->waveSpeed.y < MIN_WAVE_SPEED || config->waveSpeed.y > MAX_WAVE_SPEED) return false;
43     if (config->waveSpeed.z < MIN_WAVE_SPEED || config->waveSpeed.z > MAX_WAVE_SPEED) return false;
44     if (config->waveOffset.x < MIN_WAVE_OFFSET || config->waveOffset.x > MAX_WAVE_OFFSET) return false;
45     if (config->waveOffset.y < MIN_WAVE_OFFSET || config->waveOffset.y > MAX_WAVE_OFFSET) return false;
46     if (config->waveOffset.z < MIN_WAVE_OFFSET || config->waveOffset.z > MAX_WAVE_OFFSET) return false;
47     return true;
48 }
```

13.104.2.12 setShowLetterBoundary()

```
void raylibcpp::Text3DConfig::setShowLetterBoundary (
    bool show ) [static]
```

Definition at line 99 of file Text3DConfig.cpp.

```
99
100     _showLetterBoundary = show;
101 }
```

13.104.2.13 setShowTextBoundary()

```
void raylibcpp::Text3DConfig::setShowTextBoundary (
    bool show ) [static]
```

Definition at line 103 of file Text3DConfig.cpp.

```
103
104     _showTextBoundary = show;
105 }
```

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

- /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DConfig.hpp
- /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DConfig.cpp

13.105 raylibcpp::Text3DMeasurement Class Reference

```
#include <Text3DMeasurement.hpp>
```

Static Public Member Functions

- static Vector3 [measureText](#) (const ::Font &font, const std::string &text, float fontSize, float fontSpacing, float lineSpacing)

13.105.1 Detailed Description

Definition at line 14 of file Text3DMeasurement.hpp.

13.105.2 Member Function Documentation

13.105.2.1 measureText()

```
Vector3 raylibcpp::Text3DMeasurement::measureText (
    const ::Font & font,
    const std::string & text,
    float fontSize,
    float fontSpacing,
    float lineSpacing ) [static]
```

Definition at line 14 of file Text3DMeasurement.cpp.

```
15
16     if (text.empty()) {
17         return {0.0f, 0.25f, 0.0f};
18     }
19
20     int len = static_cast<int>(text.length());
21     int tempLen = 0;
22     int lenCounter = 0;
23     float tempTextWidth = 0.0f;
24     float scale = fontSize / static_cast<float>(font.baseSize);
25     float textHeight = scale;
26     float textWidth = 0.0f;
27
28     for (int i = 0; i < len; i++) {
29         int next = 0;
30         int letter = GetCodepoint(&text[i], &next);
31         int index = GetGlyphIndex(font, letter);
32
33         if (letter == 0x3f) next = 1;
34         i += next - 1;
35
36         if (letter != '\n') {
37             if (isWaveMarker(text, i)) {
```

```

38             i++;
39         } else {
40             lenCounter++;
41             if (font.glyphs[index].advanceX != 0) {
42                 textWidth += static_cast<float>(font.glyphs[index].advanceX) * scale;
43             } else {
44                 textWidth += static_cast<float>(font.recs[index].width + font.glyphs[index].offsetX)
45                     * scale;
46             }
47         } else {
48             tempTextWidth = std::max(tempTextWidth, textWidth);
49             lenCounter = 0;
50             textWidth = 0.0f;
51             textHeight += fontSize + lineSpacing;
52         }
53         tempLen = std::max(tempLen, lenCounter);
54     }
55     tempTextWidth = std::max(tempTextWidth, textWidth);
56     return {
57         tempTextWidth + static_cast<float>((tempLen - 1) * fontSpacing),
58         0.25f,
59         textHeight
60     };
61 }
62 }
```

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

- /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DMeasurement.hpp
- /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DMeasurement.cpp

13.106 raylibcpp::Text3DParser Class Reference

```
#include <Text3DParser.hpp>
```

Static Public Member Functions

- static bool [isWaveMarker](#) (const std::string &text, int pos)
- static int [skipWaveMarkers](#) (const std::string &text, int pos)
- static void [renderWaveText](#) (const ::Font &font, const std::string &text, Vector3 position, float fontSize, float fontSpacing, float lineSpacing, bool backface, const WaveTextConfigPtr &config, float time, Color tint)

13.106.1 Detailed Description

Definition at line 18 of file Text3DParser.hpp.

13.106.2 Member Function Documentation

13.106.2.1 [isWaveMarker\(\)](#)

```
bool raylibcpp::Text3DParser::isWaveMarker (
    const std::string & text,
    int pos ) [static]
```

Definition at line 19 of file Text3DParser.cpp.

```

19
20     return (pos < static_cast<int>(text.length()) - 1 &&
21             text[pos] == '~' && text[pos + 1] == '~');
22 }
```

13.106.2.2 [renderWaveText\(\)](#)

```
void raylibcpp::Text3DParser::renderWaveText (
    const ::Font & font,
    const std::string & text,
    Vector3 position,
```

```

    float fontSize,
    float fontSpacing,
    float lineSpacing,
    bool backface,
    const WaveTextConfigPtr & config,
    float time,
    Color tint ) [static]

```

Definition at line 31 of file Text3DParser.cpp.

```

34
35     if (!Text3DWave::isValidConfig(config)) {
36         std::cerr << "Error: WaveTextConfig is null!" << std::endl;
37         return;
38     }
39
40     int length = static_cast<int>(text.length());
41     auto renderState = std::make_shared<RenderState>();
42     renderState->position = position;
43     renderState->textOffsetY = 0.0f;
44     renderState->textOffsetX = 0.0f;
45     renderState->charIndex = 0;
46     float scale = fontSize / static_cast<float>(font.baseSize);
47     bool wave = false;
48
49     for (int i = 0; i < length; ++renderState->charIndex) {
50         int codepointByteCount = 0;
51         int codepoint = GetCodepoint(&text[i], &codepointByteCount);
52         int index = GetGlyphIndex(font, codepoint);
53
54         // Handle malformed UTF-8 sequences
55         if (codepoint == 0x3f) codepointByteCount = 1;
56
57         if (codepoint == '\n') {
58             handleWaveNewline(renderState, fontSize, lineSpacing);
59         } else if (codepoint == '~' && i + 1 < length && text[i + 1] == '~') {
60             codepointByteCount += 1;
61             wave = !wave;
62             i += codepointByteCount;
63             continue;
64         } else {
65             processWaveCharacter(font, codepoint, index, renderState,
66                                   fontSize, fontSpacing, lineSpacing, scale, backface, tint,
67                                   wave, config, time);
68         }
69         i += codepointByteCount;
70     }
71 }
72 }
```

13.106.2.3 skipWaveMarkers()

```

int raylibcpp::Text3DParser::skipWaveMarkers (
    const std::string & text,
    int pos ) [static]

```

Definition at line 24 of file Text3DParser.cpp.

```

24
25     if (isWaveMarker(text, pos)) {
26         return pos + 2;
27     }
28     return pos;
29 }
```

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

- /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DParser.hpp
- /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DParser.cpp

13.107 raylibcpp::Text3DRenderer Class Reference

```
#include <Text3DRenderer.hpp>
```

Static Public Member Functions

- static void [renderBasicText](#) (const ::Font &font, const std::string &text, Vector3 position, float fontSize, float fontSpacing, float lineSpacing, bool backface, Color tint)

13.107.1 Detailed Description

Definition at line 16 of file Text3DRenderer.hpp.

13.107.2 Member Function Documentation

13.107.2.1 renderBasicText()

```
void raylibcpp::Text3DRenderer::renderBasicText (
    const ::Font & font,
    const std::string & text,
    Vector3 position,
    float fontSize,
    float fontSpacing,
    float lineSpacing,
    bool backface,
    Color tint ) [static]
```

Definition at line 19 of file Text3DRenderer.cpp.

```
21
22     if (text.empty())
23         return;
24     int length = static_cast<int>(text.length());
25     auto renderState = std::make_shared<RenderState>();
26     renderState->position = position;
27     renderState->textOffsetY = 0.0f;
28     renderState->textOffsetX = 0.0f;
29     renderState->charIndex = 0;
30     float scale = fontSize / static_cast<float>(font.baseSize);
31
32     for (int i = 0; i < length;) {
33         int codepointByteCount = 0;
34         int codepoint = GetCodepoint(&text[i], &codepointByteCount);
35         int index = GetGlyphIndex(font, codepoint);
36         if (codepoint == 0x3f) codepointByteCount = 1;
37         processCharacter(font, codepoint, index, renderState,
38                           fontSize, fontSpacing, lineSpacing, scale, backface, tint);
39         i += codepointByteCount;
40     }
41 }
```

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

- /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/[Text3DRenderer.hpp](#)
- /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/[Text3DRenderer.cpp](#)

13.108 raylibcpp::Text3DWave Class Reference

```
#include <Text3DWave.hpp>
```

Static Public Member Functions

- static Vector3 [applyWaveEffect](#) (Vector3 basePosition, const WaveTextConfigPtr &config, float time, int charIndex)
- static WaveTextConfigPtr [createDefaultConfig](#) ()
- static bool [isValidConfig](#) (const WaveTextConfigPtr &config)

13.108.1 Detailed Description

Definition at line 15 of file Text3DWave.hpp.

13.108.2 Member Function Documentation

13.108.2.1 applyWaveEffect()

```
Vector3 raylibcpp::Text3DWave::applyWaveEffect (
    Vector3 basePosition,
    const WaveTextConfigPtr & config,
    float time,
    int charIndex ) [static]
```

Definition at line 14 of file Text3DWave.cpp.

```
15     if (!config) {
16         return basePosition;
17     }
18
19     Vector3 wavePosition = basePosition;
20
21     wavePosition.x += calculateWaveOffset(config->waveSpeed.x, config->waveOffset.x,
22                                         config->waveRange.x, time, charIndex);
23     wavePosition.y += calculateWaveOffset(config->waveSpeed.y, config->waveOffset.y,
24                                         config->waveRange.y, time, charIndex);
25     wavePosition.z += calculateWaveOffset(config->waveSpeed.z, config->waveOffset.z,
26                                         config->waveRange.z, time, charIndex);
27
28     return wavePosition;
29 }
```

13.108.2.2 createDefaultConfig()

```
WaveTextConfigPtr raylibcpp::Text3DWave::createDefaultConfig () [static]
```

Definition at line 31 of file Text3DWave.cpp.

```
31     {
32     return std::make_shared<WaveTextConfig>(WaveTextConfig{
33         .waveRange = { 0.45f, 0.45f, 0.45f },
34         .waveSpeed = { 3.0f, 3.0f, 0.5f },
35         .waveOffset = { 0.35f, 0.35f, 0.35f }
36     });
37 }
```

13.108.2.3 isValidConfig()

```
bool raylibcpp::Text3DWave::isValidConfig (
    const WaveTextConfigPtr & config ) [static]
```

Definition at line 39 of file Text3DWave.cpp.

```
39     {
40     return config != nullptr;
41 }
```

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

- /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DWave.hpp
- /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DWave.cpp

13.109 RayGUICPP::TextBox Class Reference

```
#include <RayGuiEncap.hpp>
```

Static Public Member Functions

- static bool [Draw](#) (const Rectangle &bounds, std::shared_ptr<std::string> text, int maxLength, bool editMode)

13.109.1 Detailed Description

Definition at line 95 of file RayGuiEncap.hpp.

13.109.2 Member Function Documentation

13.109.2.1 Draw()

```
static bool RayGUICPP::TextBox::Draw (
    const Rectangle & bounds,
    std::shared_ptr< std::string > text,
    int maxLength,
    bool editMode ) [inline], [static]
Definition at line 97 of file RayGuiEncap.hpp.
97
98     {
99         if (!text || maxLength <= 0)
100             return false;
101         const int bufferSize = std::min(maxLength + 1, 1024);
102         std::vector<char> buffer(bufferSize);
103         std::strncpy(buffer.data(), text->c_str(), bufferSize - 1);
104         buffer[bufferSize - 1] = '\0';
105         const int effectiveMaxLength = std::min(maxLength, bufferSize - 1);
106         bool changed = GuiTextBox(bounds, buffer.data(), effectiveMaxLength, editMode);
107         if (changed)
108             *text = buffer.data();
109     }
```

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

- /root/Desktop/Zappy/libs/RayGUICPP/include/RayGuiEncap.hpp

13.110 RayGUICPP::TextInputBox Class Reference

```
#include <RayGuiEncap.hpp>
```

Static Public Member Functions

- static int [Draw](#) (const Rectangle &bounds, const std::string &title, const std::string &message, std::shared_ptr< std::string > text, const std::string &buttons)

13.110.1 Detailed Description

Definition at line 233 of file RayGuiEncap.hpp.

13.110.2 Member Function Documentation

13.110.2.1 Draw()

```
static int RayGUICPP::TextInputBox::Draw (
    const Rectangle & bounds,
    const std::string & title,
    const std::string & message,
    std::shared_ptr< std::string > text,
    const std::string & buttons ) [inline], [static]
```

Definition at line 235 of file RayGuiEncap.hpp.

```
235
236     if (!text) return -1;
237     char buffer[1024];
238     std::strncpy(buffer, text->c_str(), sizeof(buffer));
239     buffer[sizeof(buffer)-1] = '\0';
240     int result = GuiTextInputBox(bounds, title.c_str(), message.c_str(), buttons.c_str(), buffer,
241         sizeof(buffer), nullptr);
242     *text = buffer;
243     return result;
244 }
```

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

- /root/Desktop/Zappy/libs/RayGUICPP/include/RayGuiEncap.hpp

13.111 raylibcpp::Texture Class Reference

```
#include <Texture.hpp>
```

Public Member Functions

- `Texture` (const std::string &path)
- `~Texture` ()
- void `draw` (int x, int y) const
- Texture2D `get` () const
- bool `isReady` () const

Static Public Member Functions

- static int `getNextId` ()

13.111.1 Detailed Description

Definition at line 16 of file Texture.hpp.

13.111.2 Constructor & Destructor Documentation

13.111.2.1 `Texture()`

```
raylibcpp::Texture::Texture (
    const std::string & path ) [explicit]
Definition at line 20 of file Texture.cpp.
20
21     texture = LoadTexture(path.c_str());
22 }
```

13.111.2.2 `~Texture()`

```
raylibcpp::Texture::~Texture ( )
Definition at line 24 of file Texture.cpp.
24
25     UnloadTexture(texture);
26 }
```

13.111.3 Member Function Documentation

13.111.3.1 `draw()`

```
void raylibcpp::Texture::draw (
    int x,
    int y ) const
Definition at line 28 of file Texture.cpp.
28
29     DrawTexture(texture, x, y, WHITE);
30 }
```

13.111.3.2 get()

```
Texture2D raylibcpp::Texture::get ( ) const
Definition at line 32 of file Texture.cpp.
32
33     return texture;
34 }
```

13.111.3.3 getNextId()

```
int raylibcpp::Texture::getNextId ( ) [static]
Definition at line 40 of file Texture.cpp.
40
41     return nextId++;
42 }
```

13.111.3.4 isReady()

```
bool raylibcpp::Texture::isReady ( ) const
Definition at line 36 of file Texture.cpp.
36
37     return texture.id != 0;
38 }
```

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

- /root/Desktop/Zappy/libs/RaylibCPP/texture/[Texture.hpp](#)
- /root/Desktop/Zappy/libs/RaylibCPP/texture/[Texture.cpp](#)

13.112 raylibcpp::Texture3DWrap Class Reference

```
#include <Texture.hpp>
```

Public Member Functions

- [Texture3DWrap](#) (const std::string &path)
- [~Texture3DWrap](#) ()
- void [bind](#) (int unit=0) const
- Texture2D & [get](#) ()

13.112.1 Detailed Description

Definition at line 29 of file Texture.hpp.

13.112.2 Constructor & Destructor Documentation**13.112.2.1 Texture3DWrap()**

```
raylibcpp::Texture3DWrap::Texture3DWrap (
    const std::string & path ) [explicit]
Definition at line 44 of file Texture.cpp.
44
45     texture = LoadTexture(path.c_str());
46     if (texture.id == 0)
47         throw std::runtime_error("Erreur chargement texture: " + path);
48 }
```

13.112.2.2 ~Texture3DWrap()

```
raylibcpp::Texture3DWrap::~Texture3DWrap ( )
Definition at line 50 of file Texture.cpp.
50
51     UnloadTexture(texture);
52 }
```

13.112.3 Member Function Documentation

13.112.3.1 bind()

```
void raylibcpp::Texture3DWrap::bind (
    int unit = 0 ) const
Definition at line 54 of file Texture.cpp.
54
55     rlActiveTextureSlot(unit);
56     rlEnableTexture(texture.id);
57 }
```

13.112.3.2 get()

```
Texture2D & raylibcpp::Texture3DWrap::get ( )
Definition at line 59 of file Texture.cpp.
59
60     return texture;
61 }
```

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

- /root/Desktop/Zappy/libs/RaylibCPP/texture/[Texture.hpp](#)
- /root/Desktop/Zappy/libs/RaylibCPP/texture/[Texture.cpp](#)

13.113 raylibcpp::TextureCoordinates Struct Reference

```
#include <Text3DHelper.hpp>
```

Data Fields

- float [tx](#)
- float [ty](#)
- float [tw](#)
- float [th](#)

13.113.1 Detailed Description

Definition at line 20 of file Text3DHelper.hpp.

13.113.2 Field Documentation

13.113.2.1 [th](#)

```
float raylibcpp::TextureCoordinates::th
Definition at line 21 of file Text3DHelper.hpp.
```

13.113.2.2 tw

```
float raylibcpp::TextureCoordinates::tw  
Definition at line 21 of file Text3DHelper.hpp.
```

13.113.2.3 tx

```
float raylibcpp::TextureCoordinates::tx  
Definition at line 21 of file Text3DHelper.hpp.
```

13.113.2.4 ty

```
float raylibcpp::TextureCoordinates::ty  
Definition at line 21 of file Text3DHelper.hpp.
```

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

- /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/[Text3DHelper.hpp](#)

13.114 TextureManager Class Reference

Gestionnaire de textures singleton qui implémente un pattern de pool de ressources pour éviter de charger plusieurs fois les mêmes textures.

```
#include <TextureManager.hpp>
```

Public Member Functions

- void [setGraphicsLib](#) (std::shared_ptr< [IGraphicsLib](#) > graphicsLib)
Définit la bibliothèque graphique à utiliser.
- bool [hasTextureLibSet](#) () const
Vérifie si la bibliothèque graphique a été initialisée.
- int [loadTexture](#) (const std::string &path)
Charge une texture à partir d'un chemin Si la texture existe déjà dans le cache, elle est retournée directement.
- void [drawTexture](#) (int textureId, int x, int y)
Utilise une texture préchargée pour dessin.
- void [unloadTexture](#) (int textureId)
Libère une texture spécifique.
- void [unloadAllTextures](#) ()
Libère toutes les textures en mémoire.
- bool [hasTexture](#) (const std::string &path) const
Vérifie si une texture est déjà chargée.
- int [getTextureId](#) (const std::string &path) const
Obtient l'ID d'une texture déjà chargée.
- size_t [getTextureCount](#) () const
Obtient le nombre de textures actuellement chargées.

Static Public Member Functions

- static [TextureManager](#) & [getInstance](#) ()
Obtient l'instance unique du gestionnaire de textures.

13.114.1 Detailed Description

Gestionnaire de textures singleton qui implémente un pattern de pool de ressources pour éviter de charger plusieurs fois les mêmes textures.

Definition at line 21 of file TextureManager.hpp.

13.114.2 Member Function Documentation

13.114.2.1 drawTexture()

```
void TextureManager::drawTexture (
    int textureId,
    int x,
    int y )
```

Utilise une texture préchargée pour dessin.

Parameters

<i>textureId</i>	ID de la texture à dessiner
<i>x</i>	Position X à l'écran
<i>y</i>	Position Y à l'écran

Definition at line 43 of file TextureManager.cpp.

```
43
44     std::lock_guard<std::mutex> lock(m_mutex);
45     if (!m_graphicsLib) {
46         std::cerr << "Erreur: GraphicsLib non initialisé" << std::endl;
47         return;
48     }
49     if (!m_graphicsLib->IsTextureReady(textureId)) {
50         std::cerr << "Erreur: Texture ID " << textureId << " non prête ou invalide" << std::endl;
51         return;
52     }
53     m_graphicsLib->DrawTexture2D(textureId, x, y);
54 }
```

13.114.2.2 getInstance()

```
static TextureManager& TextureManager::getInstance ( ) [inline], [static]
```

Obtient l'instance unique du gestionnaire de textures.

Returns

L'instance du [TextureManager](#)

Definition at line 27 of file TextureManager.hpp.

```
27
28     static TextureManager instance;
29     return instance;
30 }
```

13.114.2.3 getTextureCount()

```
size_t TextureManager::getTextureCount ( ) const
```

Obtient le nombre de textures actuellement chargées.

Returns

Nombre de textures en cache

Definition at line 93 of file TextureManager.cpp.

```
93
94     std::lock_guard<std::mutex> lock(m_mutex);
95     return m_pathToId.size();
96 }
```

13.114.2.4 getTextureId()

```
int TextureManager::getTextureId (
    const std::string & path ) const
```

Obtient l'ID d'une texture déjà chargée.

Parameters

<i>path</i>	Chemin d'accès à l'image de la texture
-------------	--

Returns

ID de la texture ou -1 si non trouvée

Definition at line 85 of file TextureManager.cpp.

```
85
86     std::lock_guard<std::mutex> lock(m_mutex);
87     auto it = m_pathToId.find(path);
88     if (it != m_pathToId.end())
89         return it->second;
90     return -1;
91 }
```

13.114.2.5 hasTexture()

```
bool TextureManager::hasTexture (
    const std::string & path ) const
```

Vérifie si une texture est déjà chargée.

Parameters

<i>path</i>	Chemin d'accès à l'image de la texture
-------------	--

Returns

true si la texture existe, false sinon

Definition at line 80 of file TextureManager.cpp.

```
80
81     std::lock_guard<std::mutex> lock(m_mutex);
82     return m_pathToId.find(path) != m_pathToId.end();
83 }
```

13.114.2.6 hasTextureLibSet()

```
bool TextureManager::hasTextureLibSet ( ) const [inline]
```

Vérifie si la bibliothèque graphique a été initialisée.

Returns

true si initialisée, false sinon

Definition at line 42 of file TextureManager.hpp.

```
42 { return m_graphicsLib != nullptr; }
```

13.114.2.7 loadTexture()

```
int TextureManager::loadTexture (
    const std::string & path )
```

Charge une texture à partir d'un chemin Si la texture existe déjà dans le cache, elle est retournée directement.

Parameters

<i>path</i>	Chemin d'accès à l'image de la texture
-------------	--

Returns

ID de la texture chargée

Definition at line 17 of file TextureManager.cpp.

```

17     std::lock_guard<std::mutex> lock(m_mutex);
18     auto it = m_pathToId.find(path);
19     if (it != m_pathToId.end()) {
20         return it->second;
21     }
22
23
24     if (!m_graphicsLib) {
25         std::cerr << "Erreur: GraphicsLib non initialisé dans TextureManager" << std::endl;
26         return -1;
27     }
28     try {
29         int textureId = m_graphicsLib->LoadTexture2D(path);
30         if (textureId < 0) {
31             std::cerr << "Erreur lors du chargement de la texture " << path << std::endl;
32             return -1;
33         }
34         m_pathToId[path] = textureId;
35         m_idToPath[textureId] = path;
36         return textureId;
37     } catch (const std::exception& e) {
38         std::cerr << "Erreur lors du chargement de la texture " << path << ":" << e.what() << std::endl;
39         return -1;
40     }
41 }
```

13.114.2.8 setGraphicsLib()

```
void TextureManager::setGraphicsLib (
    std::shared_ptr< IGraphicsLib

```

Définit la bibliothèque graphique à utiliser.

Parameters

<i>graphicsLib</i>	Pointeur partagé vers une implémentation de IGraphicsLib
--------------------	--

Definition at line 12 of file TextureManager.cpp.

```

12
13     std::lock_guard<std::mutex> lock(m_mutex);
14     m_graphicsLib = graphicsLib;
15 }
```

13.114.2.9 unloadAllTextures()

```
void TextureManager::unloadAllTextures ( )
```

Libère toutes les textures en mémoire.

Definition at line 70 of file TextureManager.cpp.

```

70
71     std::lock_guard<std::mutex> lock(m_mutex);
72     if (m_graphicsLib) {
73         for (const auto& [id, __] : m_idToPath)
74             m_graphicsLib->UnloadTexture2D(id);
75     }
76     m_idToPath.clear();
77     m_pathToId.clear();
78 }
```

13.114.2.10 unloadTexture()

```
void TextureManager::unloadTexture (
    int textureId )
```

Libère une texture spécifique.

Parameters

<i>texture</i> ↵ <i>Id</i>	ID de la texture à libérer
-------------------------------	----------------------------

Definition at line 56 of file TextureManager.cpp.

```
56
57     std::lock_guard<std::mutex> lock(m_mutex);
58     auto it = m_idToPath.find(textureId);
59     if (it == m_idToPath.end()) {
60         std::cerr << "Tentative de libération d'une texture inexistante (ID: " << textureId << ")" <<
61         std::endl;
62         return;
63     }
64     std::string path = it->second;
65     m_idToPath.erase(it);
66     m_pathToId.erase(path);
67     if (m_graphicsLib)
68         m_graphicsLib->UnloadTexture2D(textureId);
69 }
```

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

- /root/Desktop/Zappy/src/GUI/textureManager/TextureManager.hpp
- /root/Desktop/Zappy/src/GUI/textureManager/TextureManager.cpp

13.115 tile_s Struct Reference

```
#include <tile.h>
```

Data Fields

- int * *player_ids*
- int *player_count*
- int *player_capacity*
- int *resources* [COUNT]
- int * *egg_ids*
- int *egg_count*
- int *egg_capacity*

13.115.1 Detailed Description

Definition at line 30 of file tile.h.

13.115.2 Field Documentation

13.115.2.1 egg_capacity

```
int tile_s::egg_capacity
```

Definition at line 37 of file tile.h.

13.115.2.2 egg_count

```
int tile_s::egg_count
```

Definition at line 36 of file tile.h.

13.115.2.3 egg_ids

```
int* tile_s::egg_ids
```

Definition at line 35 of file tile.h.

13.115.2.4 player_capacity

```
int tile_s::player_capacity
```

Definition at line 33 of file tile.h.

13.115.2.5 player_count

```
int tile_s::player_count
```

Definition at line 32 of file tile.h.

13.115.2.6 player_ids

```
int* tile_s::player_ids
```

Definition at line 31 of file tile.h.

13.115.2.7 resources

```
int tile_s::resources[COUNT]
```

Definition at line 34 of file tile.h.

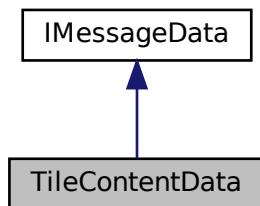
The documentation for this struct was generated from the following file:

- /root/Desktop/Zappy/src/Server/include/tile.h

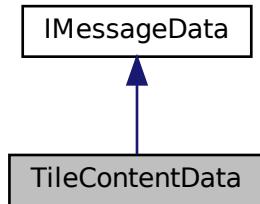
13.116 TileContentData Class Reference

```
#include <TileContentData.hpp>
```

Inheritance diagram for TileContentData:



Collaboration diagram for TileContentData:



Public Member Functions

- `TileContentData (int x, int y, int food, int linemate, int deraumere, int sibur, int mendiane, int phiras, int thystame)`
- `MessageType getType () const override`
- `int getX () const`
- `int getY () const`
- `int getFood () const`
- `int getLinemate () const`
- `int getDeraumere () const`
- `int getSibur () const`
- `int getMendiane () const`
- `int getPhiras () const`
- `int getThystame () const`
- `void setX (int value)`
- `void setY (int value)`
- `void setFood (int value)`
- `void setLinemate (int value)`
- `void setDeraumere (int value)`
- `void setSibur (int value)`
- `void setMendiane (int value)`
- `void setPhiras (int value)`
- `void setThystame (int value)`

13.116.1 Detailed Description

Definition at line 13 of file TileContentData.hpp.

13.116.2 Constructor & Destructor Documentation

13.116.2.1 TileContentData()

```
TileContentData::TileContentData (
    int x,
    int y,
    int food,
    int linemate,
    int deraumere,
```

```

    int sibur,
    int mendiane,
    int phiras,
    int thystame ) [inline]
Definition at line 15 of file TileContentData.hpp.
17     : _x(x), _y(y), _food(food), _linemate(linemate), _deraumere(deraumere),
18     _sibur(sibur), _mendiane(mendiane), _phiras(phiras), _thystame(thystame) {}

```

13.116.3 Member Function Documentation

13.116.3.1 getDeraumere()

```

int TileContentData::getDeraumere () const [inline]
Definition at line 25 of file TileContentData.hpp.
25 { return _deraumere; }

```

13.116.3.2 getFood()

```

int TileContentData::getFood () const [inline]
Definition at line 23 of file TileContentData.hpp.
23 { return _food; }

```

13.116.3.3 getLinemate()

```

int TileContentData::getLinemate () const [inline]
Definition at line 24 of file TileContentData.hpp.
24 { return _linemate; }

```

13.116.3.4 getMendiane()

```

int TileContentData::getMendiane () const [inline]
Definition at line 27 of file TileContentData.hpp.
27 { return _mendiane; }

```

13.116.3.5 getPhiras()

```

int TileContentData::getPhiras () const [inline]
Definition at line 28 of file TileContentData.hpp.
28 { return _phiras; }

```

13.116.3.6 getSibur()

```

int TileContentData::getSibur () const [inline]
Definition at line 26 of file TileContentData.hpp.
26 { return _sibur; }

```

13.116.3.7 getThystame()

```

int TileContentData::getThystame () const [inline]
Definition at line 29 of file TileContentData.hpp.
29 { return _thystame; }

```

13.116.3.8 getType()

```
MessageType TileContentData::getType () const [inline], [override], [virtual]
Implements IMessageData.
Definition at line 19 of file TileContentData.hpp.
19 { return MessageType::TileContent; }
```

13.116.3.9 getX()

```
int TileContentData::getX () const [inline]
Definition at line 21 of file TileContentData.hpp.
21 { return _x; }
```

13.116.3.10 getY()

```
int TileContentData::getY () const [inline]
Definition at line 22 of file TileContentData.hpp.
22 { return _y; }
```

13.116.3.11 setDeraumere()

```
void TileContentData::setDeraumere (
    int value ) [inline]
Definition at line 35 of file TileContentData.hpp.
35 { _derauemere = value; }
```

13.116.3.12 setFood()

```
void TileContentData::setFood (
    int value ) [inline]
Definition at line 33 of file TileContentData.hpp.
33 { _food = value; }
```

13.116.3.13 setLinemate()

```
void TileContentData::setLinemate (
    int value ) [inline]
Definition at line 34 of file TileContentData.hpp.
34 { _linemate = value; }
```

13.116.3.14 setMendiane()

```
void TileContentData::setMendiane (
    int value ) [inline]
Definition at line 37 of file TileContentData.hpp.
37 { _mendiane = value; }
```

13.116.3.15 setPhiras()

```
void TileContentData::setPhiras (
    int value ) [inline]
Definition at line 38 of file TileContentData.hpp.
38 { _phiras = value; }
```

13.116.3.16 setSibur()

```
void TileContentData::setSibur (
    int value ) [inline]
Definition at line 36 of file TileContentData.hpp.
36 { _sibur = value; }
```

13.116.3.17 setThystame()

```
void TileContentData::setThystame (
    int value ) [inline]
Definition at line 39 of file TileContentData.hpp.
39 { _thystame = value; }
```

13.116.3.18 setX()

```
void TileContentData::setX (
    int value ) [inline]
Definition at line 31 of file TileContentData.hpp.
31 { _x = value; }
```

13.116.3.19 setY()

```
void TileContentData::setY (
    int value ) [inline]
Definition at line 32 of file TileContentData.hpp.
32 { _y = value; }
```

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

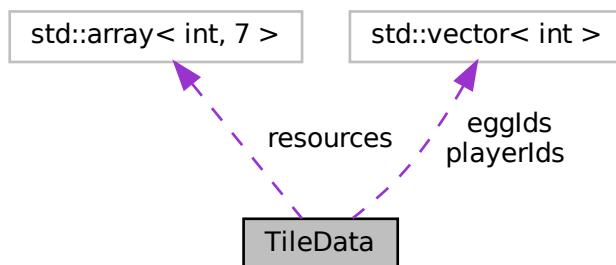
- /root/Desktop/Zappy/src/GUI/network/protocol/messageData/[TileContentData.hpp](#)

13.117 TileData Struct Reference

Structure représentant une tuile dans le contexte graphique.

```
#include <GraphicalContext.hpp>
```

Collaboration diagram for TileData:



Data Fields

- `std::array< int, 7 > resources {0}`

- std::vector< int > [playerIds](#)
- std::vector< int > [eggIds](#)
- bool [isIncantating](#) {false}

13.117.1 Detailed Description

Structure représentant une tuile dans le contexte graphique.

Definition at line 26 of file GraphicalContext.hpp.

13.117.2 Field Documentation

13.117.2.1 eggIds

std::vector<int> [TileData:::eggIds](#)

Definition at line 29 of file GraphicalContext.hpp.

13.117.2.2 isIncantating

bool [TileData:::isIncantating](#) {false}

Definition at line 30 of file GraphicalContext.hpp.

13.117.2.3 playerIds

std::vector<int> [TileData:::playerIds](#)

Definition at line 28 of file GraphicalContext.hpp.

13.117.2.4 resources

std::array<int, 7> [TileData:::resources](#) {0}

Definition at line 27 of file GraphicalContext.hpp.

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

- /root/Desktop/Zappy/src/GUI/graphicalContext/[GraphicalContext.hpp](#)

13.118 Zappy::TileRenderStrategyFactory Class Reference

```
#include <TileRenderStrategyFactory.hpp>
```

Public Member Functions

- [TileRenderStrategyFactory](#) (const std::shared_ptr< [ModelManagerAdapter](#) > &modelManager)
- std::shared_ptr< [ITileRenderStrategy](#) > [createSimpleTileStrategy](#) (const std::shared_ptr< [GraphicalContext](#) > &ctx)
- std::shared_ptr< [ITileRenderStrategy](#) > [createModelTileStrategy](#) (int modelId, const std::shared_ptr< [GraphicalContext](#) > &ctx)
- std::shared_ptr< [ITileRenderStrategy](#) > [createDetailedTileStrategy](#) (const std::shared_ptr< [GraphicalContext](#) > &ctx)

13.118.1 Detailed Description

Definition at line 19 of file TileRenderStrategyFactory.hpp.

13.118.2 Constructor & Destructor Documentation

13.118.2.1 TileRenderStrategyFactory()

```
Zappy::TileRenderStrategyFactory::TileRenderStrategyFactory (
    const std::shared_ptr< ModelManagerAdapter > & modelManager ) [explicit]
Definition at line 17 of file TileRenderStrategyFactory.cpp.
18     : modelManager(manager) {}
```

13.118.3 Member Function Documentation

13.118.3.1 createDetailedTileStrategy()

```
std::shared_ptr< ITileRenderStrategy > Zappy::TileRenderStrategyFactory::createDetailedTileStrategy (
    const std::shared_ptr< GraphicalContext > & ctx )
Definition at line 29 of file TileRenderStrategyFactory.cpp.
30     {
31     return std::make_shared<DetailedTileRenderStrategy>(ctx);
32 }
```

13.118.3.2 createModelTileStrategy()

```
std::shared_ptr< ITileRenderStrategy > Zappy::TileRenderStrategyFactory::createModelTileStrategy (
    int modelId,
    const std::shared_ptr< GraphicalContext > & ctx )
Definition at line 24 of file TileRenderStrategyFactory.cpp.
24
25     auto managerPtr = std::shared_ptr<ModelManager>(&modelManager->getManager(), [] (ModelManager*) {});
26     return std::make_shared<ModelTileRenderStrategy>(managerPtr, modelId, ctx);
27 }
```

13.118.3.3 createSimpleTileStrategy()

```
std::shared_ptr< ITileRenderStrategy > Zappy::TileRenderStrategyFactory::createSimpleTileStrategy (
    const std::shared_ptr< GraphicalContext > & ctx )
Definition at line 20 of file TileRenderStrategyFactory.cpp.
20
21     {
22     return std::make_shared<SimpleTileRenderStrategy>(ctx);
22 }
```

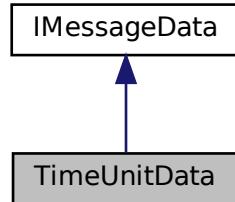
The documentation for this class was generated from the following files:

- /root/Desktop/Zappy/src/GUI/renderer/strategies/TileRenderStrategyFactory.hpp
- /root/Desktop/Zappy/src/GUI/renderer/strategies/TileRenderStrategyFactory.cpp

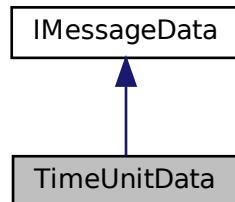
13.119 TimeUnitData Class Reference

```
#include <TimeUnitData.hpp>
```

Inheritance diagram for TimeUnitData:



Collaboration diagram for TimeUnitData:



Public Member Functions

- [TimeUnitData \(int timeUnit\)](#)
- [MessageType getType \(\) const override](#)
- [int getTimeUnit \(\) const](#)
- [void setTimeUnit \(int value\)](#)

13.119.1 Detailed Description

Definition at line 13 of file TimeUnitData.hpp.

13.119.2 Constructor & Destructor Documentation

13.119.2.1 TimeUnitData()

```
TimeUnitData::TimeUnitData (
    int timeUnit ) [inline], [explicit]
Definition at line 15 of file TimeUnitData.hpp.
```

16 : _timeUnit(timeUnit) {}

13.119.3 Member Function Documentation

13.119.3.1 getTimeUnit()

```
int TimeUnitData::getTimeUnit ( ) const [inline]
Definition at line 19 of file TimeUnitData.hpp.
19 { return _timeUnit; }
```

13.119.3.2 getType()

```
MessageType TimeUnitData::getType ( ) const [inline], [override], [virtual]
Implements IMessageData.
Definition at line 17 of file TimeUnitData.hpp.
17 { return MessageType::TimeUnit; }
```

13.119.3.3 setTimeUnit()

```
void TimeUnitData::setTimeUnit (
    int value ) [inline]
Definition at line 21 of file TimeUnitData.hpp.
21 { _timeUnit = value; }
```

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

- /root/Desktop/Zappy/src/GUI/network/protocol/messageData/[TimeUnitData.hpp](#)

13.120 RayGUICPP::Toggle Class Reference

```
#include <RayGuiEncap.hpp>
```

Static Public Member Functions

- static bool [Draw](#) (const Rectangle &bounds, const std::string &text, bool active)

13.120.1 Detailed Description

Definition at line 57 of file RayGuiEncap.hpp.

13.120.2 Member Function Documentation

13.120.2.1 Draw()

```
static bool RayGUICPP::Toggle::Draw (
    const Rectangle & bounds,
    const std::string & text,
    bool active ) [inline], [static]
Definition at line 59 of file RayGuiEncap.hpp.
59
60     bool value = active;
61     if (GuiToggle(bounds, text.c_str(), &value)) {
62         value = !active;
63     }
64     return value;
65 }
```

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

- /root/Desktop/Zappy/libs/RayGUICPP/include/[RayGuiEncap.hpp](#)

13.121 RayGUICPP::ToggleGroup Class Reference

```
#include <RayGuiEncap.hpp>
```

Static Public Member Functions

- static int [Draw](#) (const Rectangle &bounds, const std::string &text, int active)

13.121.1 Detailed Description

Definition at line 68 of file RayGuiEncap.hpp.

13.121.2 Member Function Documentation

13.121.2.1 Draw()

```
static int RayGUICPP::ToggleGroup::Draw (
    const Rectangle & bounds,
    const std::string & text,
    int active ) [inline], [static]
```

Definition at line 70 of file RayGuiEncap.hpp.

```
70
71     int value = active;
72     GuiToggleGroup(bounds, text.c_str(), &value);
73     return value;
74 }
```

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

- /root/Desktop/Zappy/libs/RayGUICPP/include/[RayGuiEncap.hpp](#)

13.122 UIRenderer Class Reference

```
#include <UIRenderer.hpp>
```

Public Member Functions

- [UIRenderer](#) ()
- [~UIRenderer](#) ()=default
- void [renderUI](#) (std::shared_ptr< [IGraphicsLib](#) > graphics, std::shared_ptr< [IGuiLib](#) > gui, std::shared_ptr< [CameraController](#) > camera)
- void [renderStatusBar](#) (std::shared_ptr< [IGuiLib](#) > gui)
- void [renderCameralInfo](#) (std::shared_ptr< [IGuiLib](#) > gui, std::shared_ptr< [CameraController](#) > camera)
- void [renderFPS](#) (std::shared_ptr< [IGraphicsLib](#) > graphics)
- bool & [showUI](#) ()
- int & [fps](#) ()

13.122.1 Detailed Description

Definition at line 16 of file UIRenderer.hpp.

13.122.2 Constructor & Destructor Documentation

13.122.2.1 UIRenderer()

```
UIRenderer::UIRenderer ( )
```

Definition at line 15 of file UIRenderer.cpp.

```
16     : m_showUI(true),
17     m_fps(60),
18     m_frameCounter(0),
19     m_windowResized(false) {
20 }
```

13.122.2.2 ~UIRenderer()

```
UIRenderer::~UIRenderer() [default]
```

13.122.3 Member Function Documentation

13.122.3.1 fps()

```
int& UIRenderer::fps() [inline]
Definition at line 27 of file UIRenderer.hpp.
27 { return m_fps; }
```

13.122.3.2 renderCameraInfo()

```
void UIRenderer::renderCameraInfo(
    std::shared_ptr< IGuiLib > gui,
    std::shared_ptr< CameraController > camera )
Definition at line 36 of file UIRenderer.cpp.
36 {
37     gui->DrawPanel(DEFAULT_WIDTH - 210, 10, 200, 80);
38     gui->DrawLabel(DEFAULT_WIDTH - 200, 20, 180, 20, "Caméra:");
39     std::string distanceStr = "Distance: " + std::to_string(static_cast<int>(camera->distance()));
40     gui->DrawLabel(DEFAULT_WIDTH - 200, 40, 180, 20, distanceStr);
41
42     std::string angleStr = "Angle: " + std::to_string(static_cast<int>(camera->angleY() * 180 / M_PI)) +
        "°";
43     gui->DrawLabel(DEFAULT_WIDTH - 200, 60, 180, 20, angleStr);
44 }
```

13.122.3.3 renderFPS()

```
void UIRenderer::renderFPS (
    std::shared_ptr< IGraphicsLib > graphics )
Definition at line 46 of file UIRenderer.cpp.
46 {
47     graphics->DrawText(("FPS: " + std::to_string(m_fps)).c_str(), 10, 10, 20, {255, 255, 0, 255});
48 }
```

13.122.3.4 renderStatusBar()

```
void UIRenderer::renderStatusBar (
    std::shared_ptr< IGuiLib > gui )
Definition at line 32 of file UIRenderer.cpp.
32 {
33     gui->DrawStatusBar(0, DEFAULT_HEIGHT - 25, DEFAULT_WIDTH, 25, "Mode Test - Contrôle de caméra avec la
        souris");
34 }
```

13.122.3.5 renderUI()

```
void UIRenderer::renderUI (
    std::shared_ptr< IGraphicsLib > graphics,
    std::shared_ptr< IGuiLib > gui,
    std::shared_ptr< CameraController > camera )
Definition at line 22 of file UIRenderer.cpp.
22 {
23     m_frameCounter++;
24     if (m_frameCounter >= 30) {
25         m_fps = m_frameCounter;
26         m_frameCounter = 0;
27     }
```

```
28     renderCameraInfo(gui, camera);
29     renderFPS(graphics);
30 }
```

13.122.3.6 showUI()

```
bool& UIRenderer::showUI ( ) [inline]
```

Definition at line 26 of file UIRenderer.hpp.

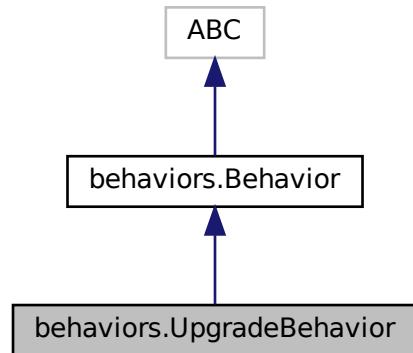
```
26 { return m_showUI; }
```

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

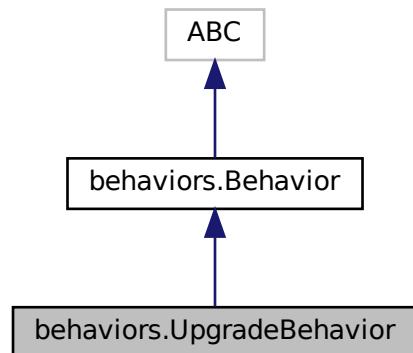
- /root/Desktop/Zappy/src/GUI/renderer/UIRenderer.hpp
- /root/Desktop/Zappy/src/GUI/renderer/UIRenderer.cpp

13.123 behaviors.UpgradeBehavior Class Reference

Inheritance diagram for behaviors.UpgradeBehavior:



Collaboration diagram for behaviors.UpgradeBehavior:



Public Member Functions

- def `execute` (self, surroundings=None, inventory=None)

Additional Inherited Members

13.123.1 Detailed Description

Definition at line 21 of file behaviors.py.

13.123.2 Member Function Documentation

13.123.2.1 execute()

```
def behaviors.UpgradeBehavior.execute (
    self,
    surroundings = None,
    inventory = None )
```

Reimplemented from [behaviors.Behavior](#).

Definition at line 22 of file behaviors.py.

```
22     def execute(self, surroundings=None, inventory=None):
23         if not surroundings or not inventory:
24             print("UpgradeBehavior: Surroundings or inventory is None.")
25             return
26
27         inventory_dict = zappy.parse_inventory(inventory)
28         upgrade_info = upgrades.upgrades.get(self.agent.level, {})
29
30         if not upgrade_info:
31             print(f"UpgradeBehavior: No upgrade defined for level {self.agent.level}.")
32             return
33         if not inventory_dict:
34             print("UpgradeBehavior: Inventory is empty or not properly parsed.")
35             return
36
37         upgrade_cost = upgrade_info.get("cost", {})
38
39         for resource, amount in upgrade_cost.items():
40             if resource == "players":
41                 if zappy.how_much_of_item_here(surroundings, "player") < amount:
42                     self.agent.send_command("Broadcast " + encryption.encrypt_message("HELP! Upgrade: 2"))
43                     return
44                 continue
45             elif inventory_dict.get(resource, 0) < amount:
46                 return
47
48         self.agent.send_command("Incantation")
```

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

- /root/Desktop/Zappy/src/AI/agent/[behaviors.py](#)

13.124 raylibcpp::Utils Class Reference

```
#include <Utils.hpp>
```

Static Public Member Functions

- static Color `hexToColor` (const std::string &hex)

13.124.1 Detailed Description

Definition at line 15 of file Utils.hpp.

13.124.2 Member Function Documentation

13.124.2.1 hexToColor()

```
Color raylibcpp::Utils::hexToColor (
    const std::string & hex ) [static]
Definition at line 15 of file Utils.cpp.
15
16     unsigned int rgb = 0;
17     std::stringstream ss;
18     ss << std::hex << hex.substr(1);
19     ss >> rgb;
20     return {(unsigned char)((rgb >> 16) & 0xFF),
21             (unsigned char)((rgb >> 8) & 0xFF), (unsigned char)(rgb & 0xFF), 255};
22 }
```

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

- /root/Desktop/Zappy/libs/RaylibCPP/utils/Utils.hpp
- /root/Desktop/Zappy/libs/RaylibCPP/utils/Utils.cpp

13.125 RayGUICPP::ValueBox Class Reference

```
#include <RayGuiEncap.hpp>
```

Static Public Member Functions

- static bool [Draw](#) (const Rectangle &bounds, std::shared_ptr< int > value, int minValue, int maxValue, bool editMode)

13.125.1 Detailed Description

Definition at line 112 of file RayGuiEncap.hpp.

13.125.2 Member Function Documentation

13.125.2.1 Draw()

```
static bool RayGUICPP::ValueBox::Draw (
    const Rectangle & bounds,
    std::shared_ptr< int > value,
    int minValue,
    int maxValue,
    bool editMode ) [inline], [static]
```

Definition at line 114 of file RayGuiEncap.hpp.

```
114
115     if (!value) return false;
116     int temp = *value;
117     bool changed = GuiValueBox(bounds, nullptr, &temp, minValue, maxValue, editMode);
118     if (changed) *value = temp;
119     return changed;
120 }
```

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

- /root/Desktop/Zappy/libs/RayGUICPP/include/[RayGuiEncap.hpp](#)

13.126 ZappyTypes::Vector2 Struct Reference

```
#include <Common.hpp>
```

Data Fields

- float [x](#)
- float [y](#)

13.126.1 Detailed Description

Definition at line 12 of file Common.hpp.

13.126.2 Field Documentation

13.126.2.1 x

```
float ZappyTypes::Vector2::x
```

Definition at line 13 of file Common.hpp.

13.126.2.2 y

```
float ZappyTypes::Vector2::y
```

Definition at line 14 of file Common.hpp.

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

- /root/Desktop/Zappy/src/Shared/[Common.hpp](#)

13.127 ZappyTypes::Vector3 Struct Reference

```
#include <Common.hpp>
```

Data Fields

- float [x](#)
- float [y](#)
- float [z](#)

13.127.1 Detailed Description

Definition at line 17 of file Common.hpp.

13.127.2 Field Documentation

13.127.2.1 x

```
float ZappyTypes::Vector3::x
```

Definition at line 18 of file Common.hpp.

13.127.2.2 y

```
float ZappyTypes::Vector3::y
```

Definition at line 19 of file Common.hpp.

13.127.2.3 z

```
float ZappyTypes::Vector3::z
Definition at line 20 of file Common.hpp.
```

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

- /root/Desktop/Zappy/src/Shared/Common.hpp

13.128 raylibcpp::Vector3Wrap Class Reference

```
#include <Utils.hpp>
```

Public Member Functions

- [Vector3Wrap](#) (float x=0, float y=0, float z=0)
- Vector3 & [get](#) ()
- void [set](#) (float x, float y, float z)

Static Public Member Functions

- static Vector3 [add](#) (const Vector3 &a, const Vector3 &b)
- static Vector3 [sub](#) (const Vector3 &a, const Vector3 &b)
- static Vector3 [scale](#) (const Vector3 &v, float s)
- static Vector3 [normalize](#) (const Vector3 &v)
- static Vector3 [cross](#) (const Vector3 &a, const Vector3 &b)
- static Vector3 [transform](#) (const Vector3 &v, const Matrix &m)

13.128.1 Detailed Description

Definition at line 20 of file Utils.hpp.

13.128.2 Constructor & Destructor Documentation

13.128.2.1 Vector3Wrap()

```
raylibcpp::Vector3Wrap::Vector3Wrap (
    float x = 0,
    float y = 0,
    float z = 0 ) [explicit]
```

Definition at line 24 of file Utils.cpp.

```
24
25     vec.x = x;
26     vec.y = y;
27     vec.z = z;
28 }
```

13.128.3 Member Function Documentation

13.128.3.1 add()

```
Vector3 raylibcpp::Vector3Wrap::add (
    const Vector3 & a,
    const Vector3 & b ) [static]
```

Definition at line 40 of file Utils.cpp.

```
40
41     return Vector3Add(a, b);
42 }
```

13.128.3.2 cross()

```
Vector3 raylibcpp::Vector3Wrap::cross (
    const Vector3 & a,
    const Vector3 & b) [static]
```

Definition at line 56 of file Utils.cpp.

```
56
57     return Vector3CrossProduct(a, b);
58 }
```

{

13.128.3.3 get()

```
Vector3 & raylibcpp::Vector3Wrap::get ()
```

Definition at line 30 of file Utils.cpp.

```
30
31     return vec;
32 }
```

{

13.128.3.4 normalize()

```
Vector3 raylibcpp::Vector3Wrap::normalize (
    const Vector3 & v) [static]
```

Definition at line 52 of file Utils.cpp.

```
52
53     return Vector3Normalize(v);
54 }
```

{

13.128.3.5 scale()

```
Vector3 raylibcpp::Vector3Wrap::scale (
    const Vector3 & v,
    float s) [static]
```

Definition at line 48 of file Utils.cpp.

```
48
49     return Vector3Scale(v, s);
50 }
```

{

13.128.3.6 set()

```
void raylibcpp::Vector3Wrap::set (
    float x,
    float y,
    float z)
```

Definition at line 34 of file Utils.cpp.

```
34
35     vec.x = x;
36     vec.y = y;
37     vec.z = z;
38 }
```

{

13.128.3.7 sub()

```
Vector3 raylibcpp::Vector3Wrap::sub (
    const Vector3 & a,
    const Vector3 & b) [static]
```

Definition at line 44 of file Utils.cpp.

```
44
45     return Vector3Subtract(a, b);
46 }
```

{

13.128.3.8 transform()

```
Vector3 raylibcpp::Vector3Wrap::transform (
    const Vector3 & v,
    const Matrix & m ) [static]
Definition at line 60 of file Utils.cpp.
```

60 {
61 return Vector3Transform(v, m);
62 }

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

- /root/Desktop/Zappy/libs/RaylibCPP/utils/[Utils.hpp](#)
- /root/Desktop/Zappy/libs/RaylibCPP/utils/[Utils.cpp](#)

13.129 raylibcpp::WaveTextConfig Struct Reference

```
#include <Text3DCore.hpp>
```

Data Fields

- Vector3 [waveRange](#)
- Vector3 [waveSpeed](#)
- Vector3 [waveOffset](#)

13.129.1 Detailed Description

Definition at line 19 of file Text3DCore.hpp.

13.129.2 Field Documentation

13.129.2.1 [waveOffset](#)

```
Vector3 raylibcpp::WaveTextConfig::waveOffset
Definition at line 22 of file Text3DCore.hpp.
```

13.129.2.2 [waveRange](#)

```
Vector3 raylibcpp::WaveTextConfig::waveRange
Definition at line 20 of file Text3DCore.hpp.
```

13.129.2.3 [waveSpeed](#)

```
Vector3 raylibcpp::WaveTextConfig::waveSpeed
Definition at line 21 of file Text3DCore.hpp.
The documentation for this struct was generated from the following file:
```

- /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/[Text3DCore.hpp](#)

13.130 raylibcpp::Window Class Reference

```
#include <Window.hpp>
```

Public Member Functions

- `Window` (int width, int height, const std::string &title)
- `~Window` ()
- bool `shouldClose` () const
- void `beginDrawing` () const
- void `endDrawing` () const
- void `clear` (Color color=RAYWHITE) const
- void `setFps` (int fps) const

13.130.1 Detailed Description

Definition at line 14 of file Window.hpp.

13.130.2 Constructor & Destructor Documentation

13.130.2.1 `Window()`

```
raylibcpp::Window::Window (
    int width,
    int height,
    const std::string & title )
```

Definition at line 13 of file Window.cpp.

```
13
14     InitWindow(width, height, title.c_str());
```

{

13.130.2.2 `~Window()`

```
raylibcpp::Window::~Window ( )
```

Definition at line 17 of file Window.cpp.

```
17
18     {
19 }
```

13.130.3 Member Function Documentation

13.130.3.1 `beginDrawing()`

```
void raylibcpp::Window::beginDrawing ( ) const
```

Definition at line 25 of file Window.cpp.

```
25
26     BeginDrawing();
27 }
```

{

13.130.3.2 `clear()`

```
void raylibcpp::Window::clear (
    Color color = RAYWHITE ) const
```

Definition at line 33 of file Window.cpp.

```
33
34     ClearBackground(color);
35 }
```

13.130.3.3 endDrawing()

```
void raylibcpp::Window::endDrawing ( ) const
Definition at line 29 of file Window.cpp.
29
30     EndDrawing();
31 }
```

13.130.3.4 setFps()

```
void raylibcpp::Window::setFps (
    int fps ) const
Definition at line 37 of file Window.cpp.
37
38     SetTargetFPS(fps);
39 }
```

13.130.3.5 shouldClose()

```
bool raylibcpp::Window::shouldClose ( ) const
Definition at line 21 of file Window.cpp.
21
22     return WindowShouldClose();
23 }
```

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

- /root/Desktop/Zappy/libs/RaylibCPP/window/[Window.hpp](#)
- /root/Desktop/Zappy/libs/RaylibCPP/window/[Window.cpp](#)

13.131 RayGUICPP::WindowBox Class Reference

```
#include <RayGuiEncap.hpp>
```

Static Public Member Functions

- static bool [Draw](#) (const Rectangle &bounds, const std::string &title)

13.131.1 Detailed Description

Definition at line 159 of file RayGuiEncap.hpp.

13.131.2 Member Function Documentation**13.131.2.1 Draw()**

```
static bool RayGUICPP::WindowBox::Draw (
    const Rectangle & bounds,
    const std::string & title ) [inline], [static]
Definition at line 161 of file RayGuiEncap.hpp.
161
162     return GuiWindowBox(bounds, title.c_str());
163 }
```

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

- /root/Desktop/Zappy/libs/RayGUICPP/include/[RayGuiEncap.hpp](#)

Chapter 14

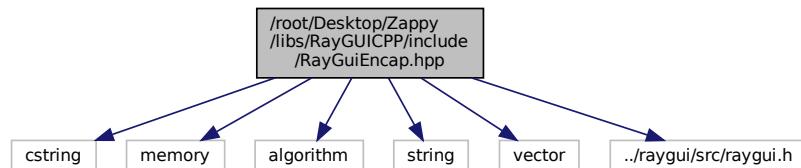
File Documentation

14.1 /root/Desktop/Zappy/docs/index.md File Reference

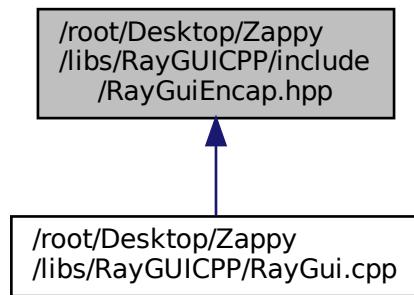
14.2 /root/Desktop/Zappy/docs/protocol_server_ai.md File Reference

14.3 /root/Desktop/Zappy/libs/RayGUICPP/include/RayGuiEncap.hpp File Reference

```
#include <cstring>
#include <memory>
#include <algorithm>
#include <string>
#include <vector>
#include "../raygui/src/raygui.h"
Include dependency graph for RayGuiEncap.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

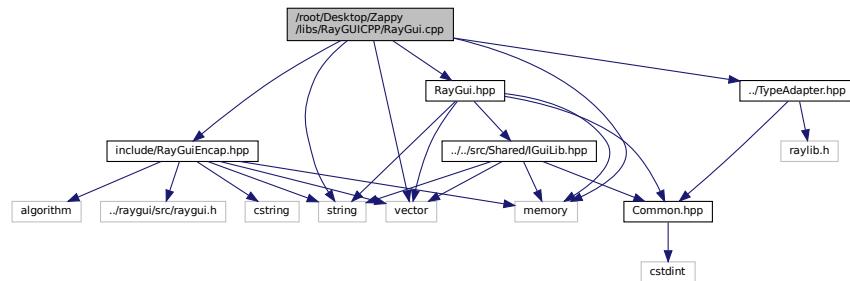
- class [RayGUICPP::RayGui](#)
- class [RayGUICPP::Button](#)
- class [RayGUICPP::Label](#)
- class [RayGUICPP::CheckBox](#)
- class [RayGUICPP::Toggle](#)
- class [RayGUICPP::ToggleGroup](#)
- class [RayGUICPP::ComboBox](#)
- class [RayGUICPP::DropdownBox](#)
- class [RayGUICPP::TextBox](#)
- class [RayGUICPP::ValueBox](#)
- class [RayGUICPP::Spinner](#)
- class [RayGUICPP::Slider](#)
- class [RayGUICPP::ProgressBar](#)
- class [RayGUICPP::StatusBar](#)
- class [RayGUICPP::WindowBox](#)
- class [RayGUICPP::GroupBox](#)
- class [RayGUICPP::Panel](#)
- class [RayGUICPP::ScrollPane](#)
- class [RayGUICPP::TabBar](#)
- class [RayGUICPP::ListView](#)
- class [RayGUICPP::ColorPicker](#)
- class [RayGUICPP::MessageBox](#)
- class [RayGUICPP::TextInputBox](#)
- class [RayGUICPP::Icon](#)

Namespaces

- [RayGUICPP](#)

14.4 /root/Desktop/Zappy/libs/RayGUICPP/RayGui.cpp File Reference

```
#include <vector>
#include <string>
#include <memory>
#include "RayGui.hpp"
#include "../TypeAdapter.hpp"
#include "include/RayGuiEncap.hpp"
Include dependency graph for RayGui.cpp:
```



Functions

- std::shared_ptr<IGuiLib> [createGuiLib\(\)](#)

14.4.1 Function Documentation

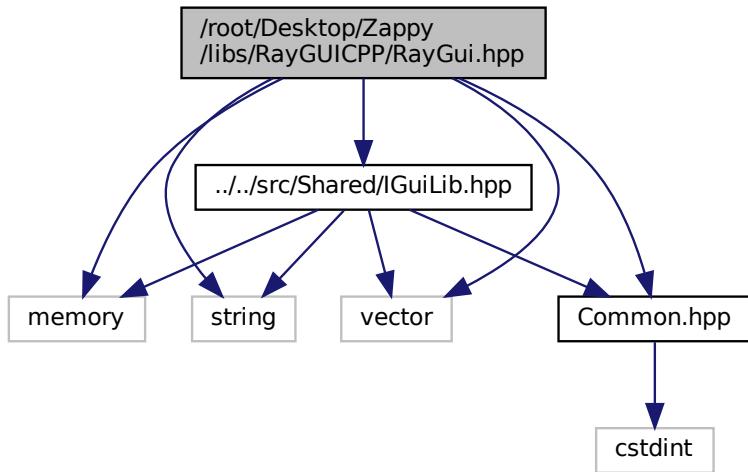
14.4.1.1 [createGuiLib\(\)](#)

```
std::shared_ptr<IGuiLib> createGuiLib ( )
Definition at line 105 of file RayGui.cpp.
105      static std::shared_ptr<RayGui> instance = std::make_shared<RayGui>();
106      return instance;
107  }
```

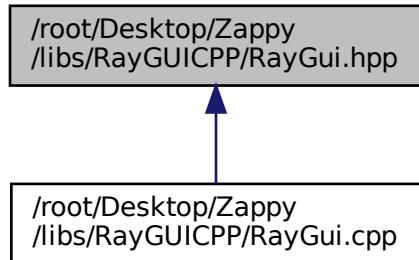
14.5 /root/Desktop/Zappy/libs/RayGUICPP/RayGui.hpp File Reference

```
#include <memory>
#include <string>
#include <vector>
#include "../../src/Shared/IGuiLib.hpp"
#include "../../src/Shared/Common.hpp"
```

Include dependency graph for RayGui.hpp:



This graph shows which files directly or indirectly include this file:



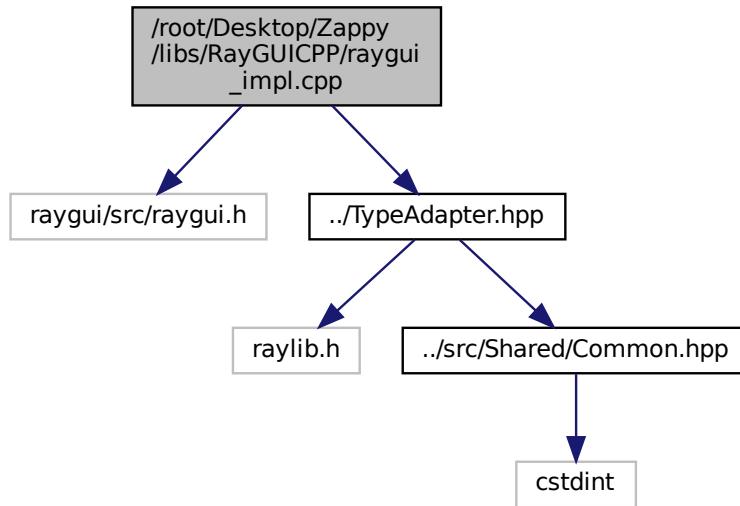
Data Structures

- class [RayGui](#)

14.6 /root/Desktop/Zappy/libs/RayGUICPP/raygui_impl.cpp File Reference

```
#include "raygui/src/raygui.h"
#include "../TypeAdapter.hpp"
```

Include dependency graph for raygui_impl.cpp:



Macros

- #define RAYLIB_VECTOR2_DEFINED
- #define RAYLIB_VECTOR3_DEFINED
- #define RAYLIB_COLOR_DEFINED
- #define RAYLIB_RECTANGLE_DEFINED
- #define RAYGUI_IMPLEMENTATION

14.6.1 Macro Definition Documentation

14.6.1.1 RAYGUI_IMPLEMENTATION

```
#define RAYGUI_IMPLEMENTATION
```

Definition at line 15 of file raygui_impl.cpp.

14.6.1.2 RAYLIB_COLOR_DEFINED

```
#define RAYLIB_COLOR_DEFINED
```

Definition at line 11 of file raygui_impl.cpp.

14.6.1.3 RAYLIB_RECTANGLE_DEFINED

```
#define RAYLIB_RECTANGLE_DEFINED
```

Definition at line 12 of file raygui_impl.cpp.

14.6.1.4 RAYLIB_VECTOR2_DEFINED

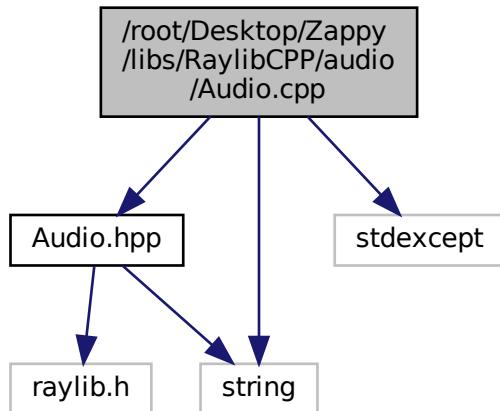
```
#define RAYLIB_VECTOR2_DEFINED  
Definition at line 9 of file raygui_impl.cpp.
```

14.6.1.5 RAYLIB_VECTOR3_DEFINED

```
#define RAYLIB_VECTOR3_DEFINED  
Definition at line 10 of file raygui_impl.cpp.
```

14.7 /root/Desktop/Zappy/libs/RaylibCPP/audio/Audio.cpp File Reference

```
#include "Audio.hpp"  
#include <stdexcept>  
#include <string>  
Include dependency graph for Audio.cpp:
```



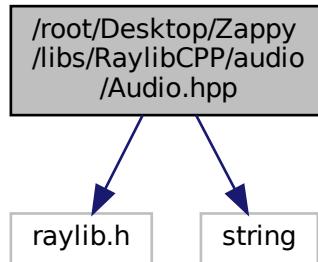
Namespaces

- [raylibcpp](#)

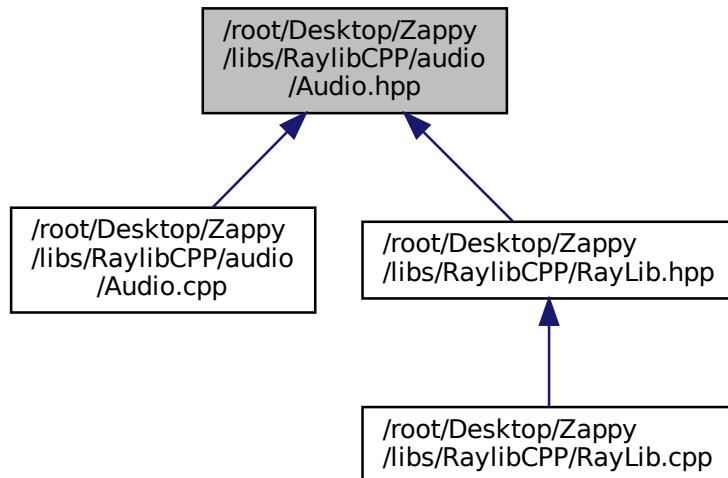
14.8 /root/Desktop/Zappy/libs/RaylibCPP/audio/Audio.hpp File Reference

```
#include <raylib.h>  
#include <string>
```

Include dependency graph for Audio.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class `raylibcpp::Audio`
- class `raylibcpp::SoundWrap`
- class `raylibcpp::MusicWrap`

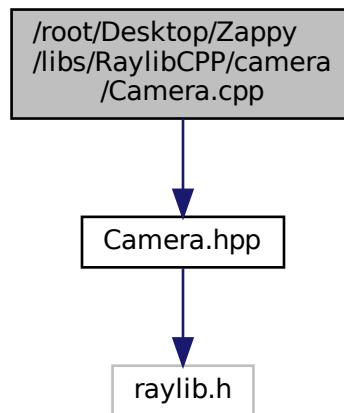
Namespaces

- `raylibcpp`

14.9 /root/Desktop/Zappy/libs/RaylibCPP/camera/Camera.cpp File Reference

```
#include "Camera.hpp"
```

Include dependency graph for Camera.cpp:



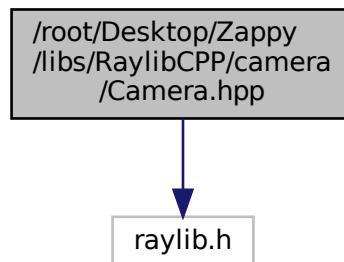
Namespaces

- [raylibcpp](#)

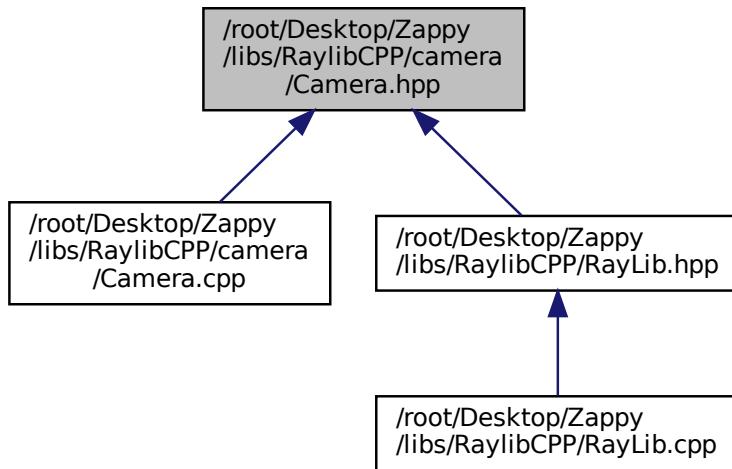
14.10 /root/Desktop/Zappy/libs/RaylibCPP/camera/Camera.hpp File Reference

```
#include <raylib.h>
```

Include dependency graph for Camera.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [raylibcpp::Camera2DWrap](#)
- class [raylibcpp::Camera3DWrap](#)

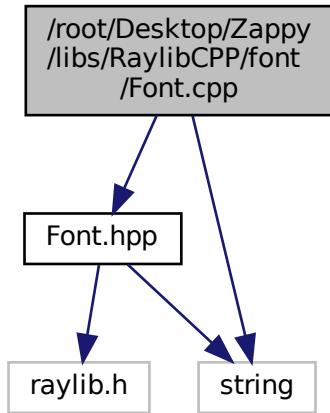
Namespaces

- [raylibcpp](#)

14.11 /root/Desktop/Zappy/libs/RaylibCPP/font/Font.cpp File Reference

```
#include "Font.hpp"
#include <string>
```

Include dependency graph for Font.cpp:

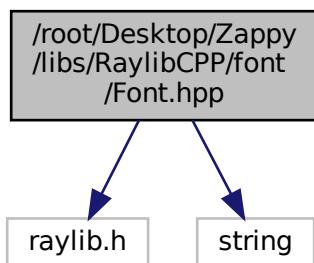


Namespaces

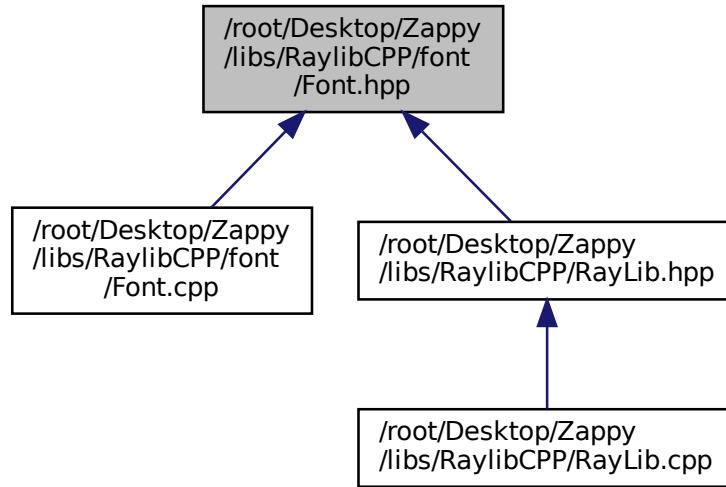
- [raylibcpp](#)

14.12 /root/Desktop/Zappy/libs/RaylibCPP/font/Font.hpp File Reference

```
#include <raylib.h>
#include <string>
Include dependency graph for Font.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class `raylibcpp::Font`

Namespaces

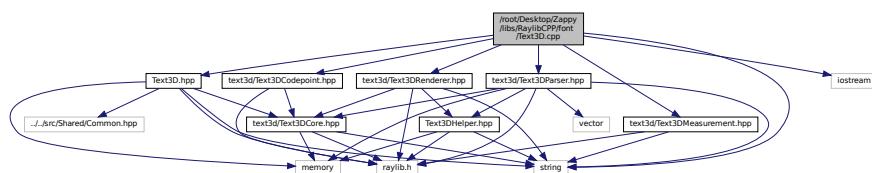
- `raylibcpp`

14.13 /root/Desktop/Zappy/libs/RaylibCPP/font/Text3D.cpp File Reference

```

#include "Text3D.hpp"
#include <iostream>
#include <string>
#include "text3d/Text3DCodepoint.hpp"
#include "text3d/Text3DRenderer.hpp"
#include "text3d/Text3DParser.hpp"
#include "text3d/Text3DMeasurement.hpp"
  
```

Include dependency graph for `Text3D.cpp`:

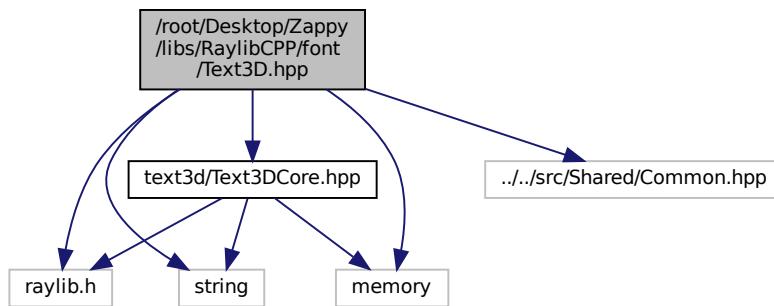


Namespaces

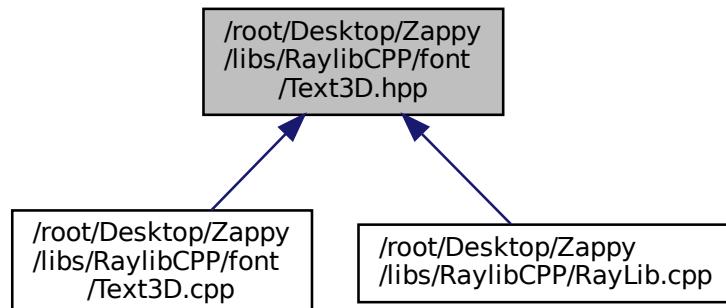
- `raylibcpp`

14.14 /root/Desktop/Zappy/libs/RaylibCPP/font/Text3D.hpp File Reference

```
#include <raylib.h>
#include <string>
#include <memory>
#include "../../src/Shared/Common.hpp"
#include "text3d/Text3DCore.hpp"
Include dependency graph for Text3D.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [raylibcpp::Text3D](#)

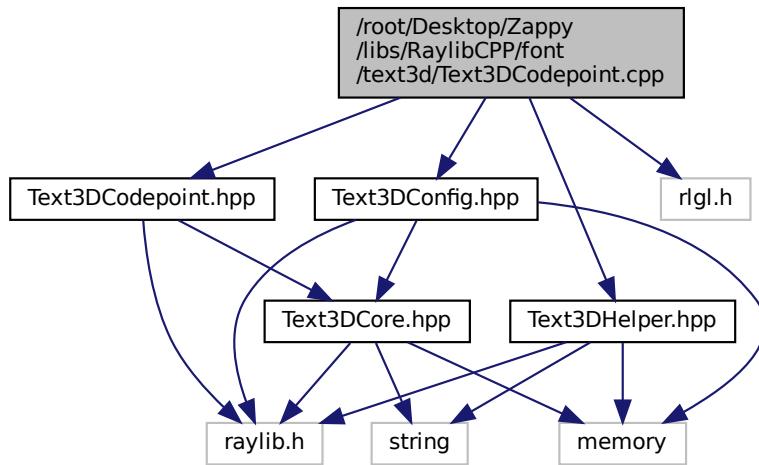
Namespaces

- [raylibcpp](#)

14.15 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DCodepoint.cpp File Reference

```
#include "Text3DCodepoint.hpp"
#include "Text3DHelper.hpp"
#include "Text3DConfig.hpp"
#include "rlgl.h"

Include dependency graph for Text3DCodepoint.cpp:
```



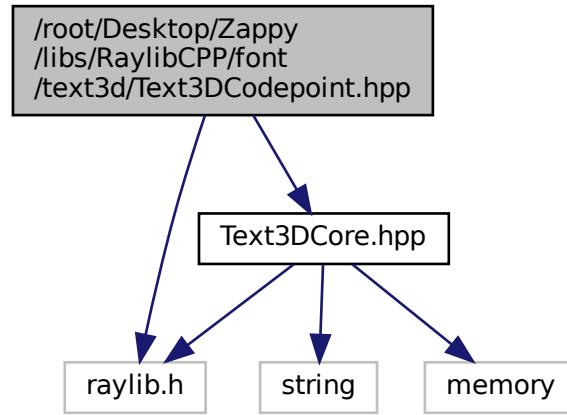
Namespaces

- [raylibcpp](#)

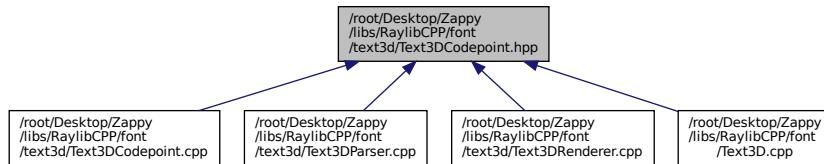
14.16 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DCodepoint.hpp File Reference

```
#include <raylib.h>
#include "Text3DCore.hpp"
```

Include dependency graph for Text3DCodepoint.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [raylibcpp::Text3DCodepoint](#)

Namespaces

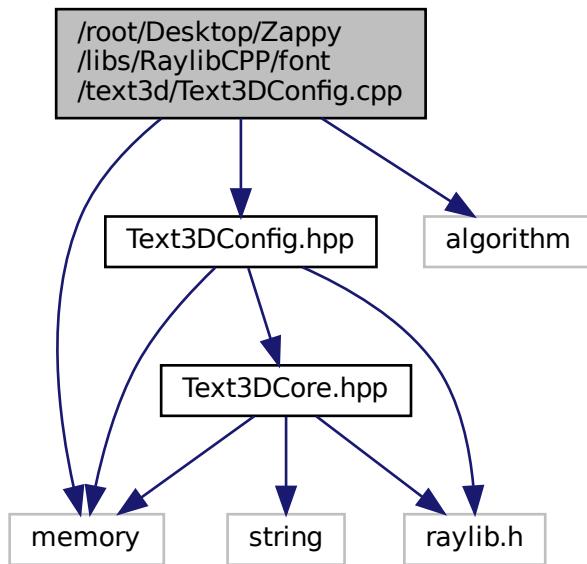
- [raylibcpp](#)

14.17 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DConfig.cpp

File Reference

```
#include "Text3DConfig.hpp"
#include <algorithm>
#include <memory>
```

Include dependency graph for Text3DConfig.cpp:



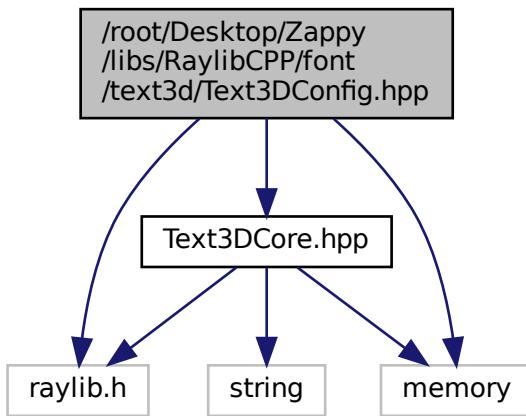
Namespaces

- [raylibcpp](#)

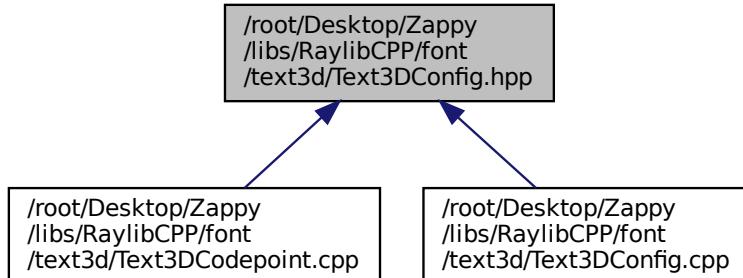
14.18 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DConfig.hpp File Reference

```
#include <raylib.h>
#include <memory>
#include "Text3DCore.hpp"
```

Include dependency graph for Text3DConfig.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [raylibcpp::Text3DConfig](#)

Namespaces

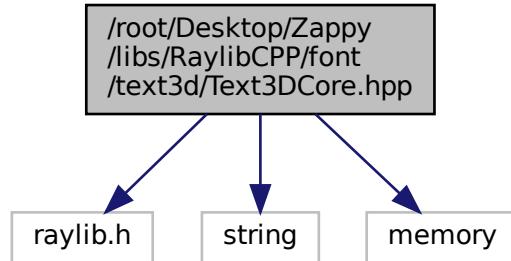
- [raylibcpp](#)

14.19 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DCore.hpp

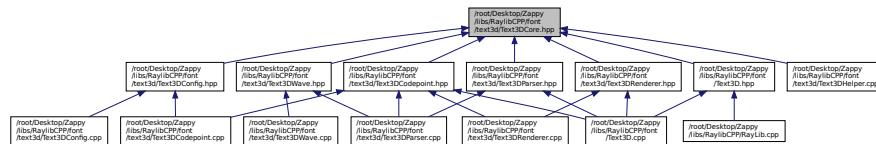
File Reference

```
#include <raylib.h>
#include <string>
```

```
#include <memory>
Include dependency graph for Text3DCore.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- struct `raylibcpp::WaveTextConfig`

Namespaces

- `raylibcpp`

Macros

- `#define LETTER_BOUNDARY_SIZE 0.25f`
- `#define TEXT_MAX_LAYERS 32`
- `#define LETTER_BOUNDARY_COLOR VIOLET`

Typedefs

- using `raylibcpp::WaveTextConfigPtr = std::shared_ptr< WaveTextConfig >`
- using `raylibcpp::WaveTextConfigUPtr = std::unique_ptr< WaveTextConfig >`

14.19.1 Macro Definition Documentation

14.19.1.1 LETTER_BOUNDARY_COLOR

```
#define LETTER_BOUNDARY_COLOR VIOLET
Definition at line 15 of file Text3DCore.hpp.
```

14.19.1.2 LETTER_BOUNDARY_SIZE

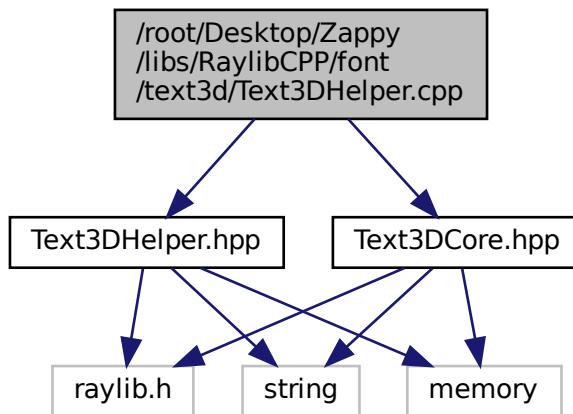
```
#define LETTER_BOUNDARY_SIZE 0.25f
Definition at line 13 of file Text3DCore.hpp.
```

14.19.1.3 TEXT_MAX_LAYERS

```
#define TEXT_MAX_LAYERS 32
Definition at line 14 of file Text3DCore.hpp.
```

14.20 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DHelper.cpp File Reference

```
#include "Text3DHelper.hpp"
#include "Text3DCore.hpp"
Include dependency graph for Text3DHelper.hpp:
```



Namespaces

- `raylibcpp`
- `raylibcpp::Text3DHelper`

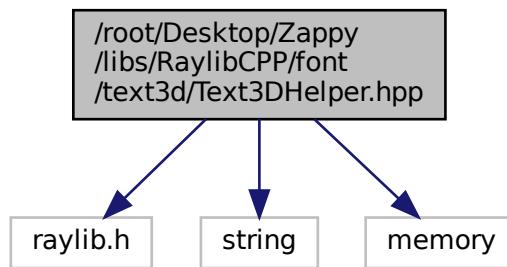
Functions

- int `raylibcpp::Text3DHelper::getGlyphIndex` (const ::Font &font, int codepoint)
- float `raylibcpp::Text3DHelper::calculateScale` (const ::Font &font, float fontSize)
- Vector3 `raylibcpp::Text3DHelper::calculateGlyphPosition` (const ::Font &font, int index, Vector3 basePosition, float scale)
- Rectangle `raylibcpp::Text3DHelper::calculateSourceRectangle` (const ::Font &font, int index)
- GlyphDimensions `raylibcpp::Text3DHelper::calculateGlyphDimensions` (const ::Font &font, int index, float scale)
- bool `raylibcpp::Text3DHelper::isRenderableCharacter` (int codepoint)
- float `raylibcpp::Text3DHelper::calculateAdvanceX` (const ::Font &font, int index, float scale, float fontSpacing)
- TextureCoordinates `raylibcpp::Text3DHelper::calculateTextureCoordinates` (const ::Font &font, Rectangle srcRec)

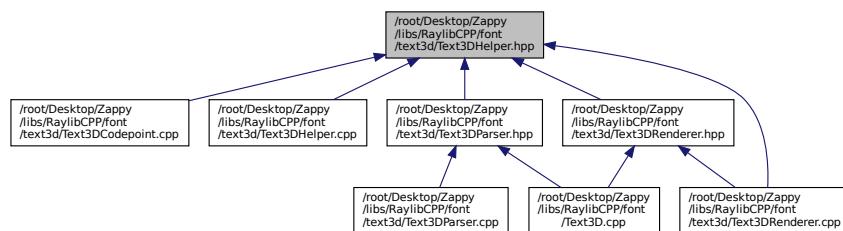
- bool `raylibcpp::Text3DHelper::isValidFont` (const ::Font &font)
- bool `raylibcpp::Text3DHelper::isValidRenderParams` (float fontSize, float fontSpacing, float lineSpacing)
- bool `raylibcpp::Text3DHelper::isWhitespace` (int codepoint)
- bool `raylibcpp::Text3DHelper::isNewline` (int codepoint)

14.21 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DHelper.hpp File Reference

```
#include <raylib.h>
#include <string>
#include <memory>
Include dependency graph for Text3DHelper.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- struct `raylibcpp::GlyphDimensions`
- struct `raylibcpp::TextureCoordinates`
- struct `raylibcpp::RenderState`

Namespaces

- `raylibcpp`
- `raylibcpp::Text3DHelper`

Typedefs

- using `raylibcpp::RenderStatePtr = std::shared_ptr< RenderState >`

Functions

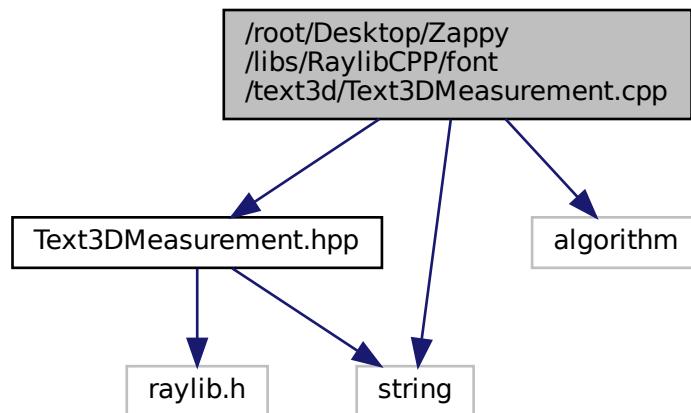
- int `raylibcpp::Text3DHelper::getGlyphIndex` (const ::Font &font, int codepoint)
- float `raylibcpp::Text3DHelper::calculateScale` (const ::Font &font, float fontSize)
- Vector3 `raylibcpp::Text3DHelper::calculateGlyphPosition` (const ::Font &font, int index, Vector3 basePosition, float scale)
- Rectangle `raylibcpp::Text3DHelper::calculateSourceRectangle` (const ::Font &font, int index)
- GlyphDimensions `raylibcpp::Text3DHelper::calculateGlyphDimensions` (const ::Font &font, int index, float scale)
- bool `raylibcpp::Text3DHelper::isRenderableCharacter` (int codepoint)
- float `raylibcpp::Text3DHelper::calculateAdvanceX` (const ::Font &font, int index, float scale, float fontSpacing)
- TextureCoordinates `raylibcpp::Text3DHelper::calculateTextureCoordinates` (const ::Font &font, Rectangle srcRec)
- bool `raylibcpp::Text3DHelper::isValidFont` (const ::Font &font)
- bool `raylibcpp::Text3DHelper::isValidRenderParams` (float fontSize, float fontSpacing, float lineSpacing)
- bool `raylibcpp::Text3DHelper::isWhitespace` (int codepoint)
- bool `raylibcpp::Text3DHelper::isNewline` (int codepoint)

Variables

- static constexpr float `raylibcpp::Text3DHelper::MIN_FONT_SIZE` = 0.1f
- static constexpr float `raylibcpp::Text3DHelper::MAX_FONT_SIZE` = 1000.0f
- static constexpr float `raylibcpp::Text3DHelper::MIN_SPACING` = -100.0f
- static constexpr float `raylibcpp::Text3DHelper::MAX_SPACING` = 1000.0f

14.22 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DMeasurement.cpp File Reference

```
#include "Text3DMeasurement.hpp"
#include <algorithm>
#include <string>
Include dependency graph for Text3DMeasurement.cpp:
```

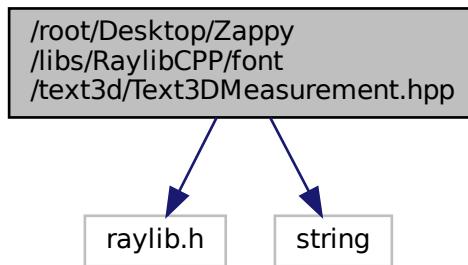


Namespaces

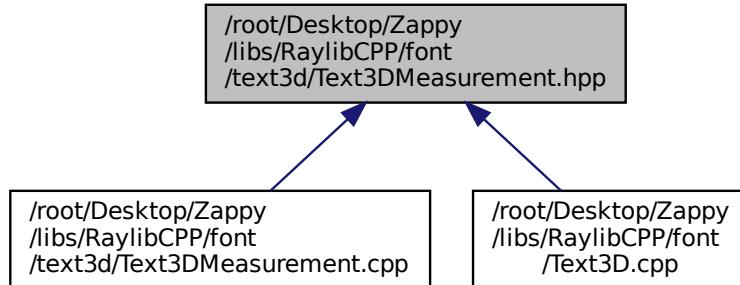
- `raylibcpp`

14.23 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DMeasurement.hpp File Reference

```
#include <raylib.h>
#include <string>
Include dependency graph for Text3DMeasurement.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [raylibcpp::Text3DMeasurement](#)

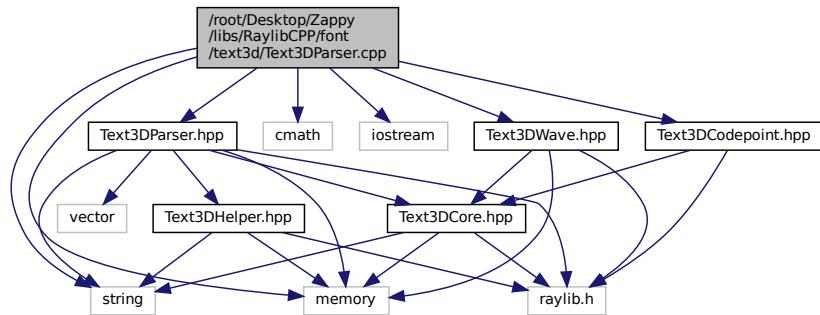
Namespaces

- [raylibcpp](#)

14.24 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DParser.cpp File Reference

```
#include "Text3DParser.hpp"
#include <cmath>
#include <iostream>
```

```
#include <string>
#include <memory>
#include "Text3DCodepoint.hpp"
#include "Text3DWave.hpp"
Include dependency graph for Text3DParser.cpp:
```

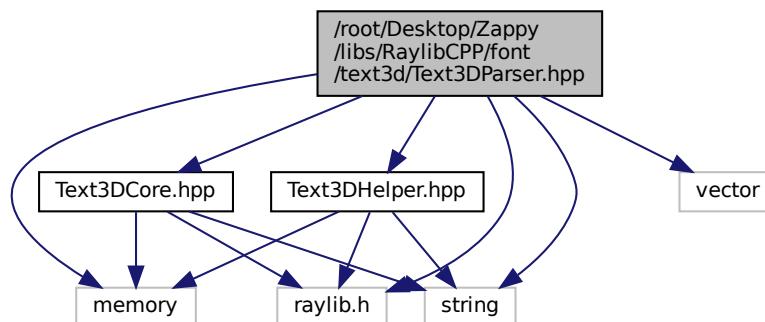


Namespaces

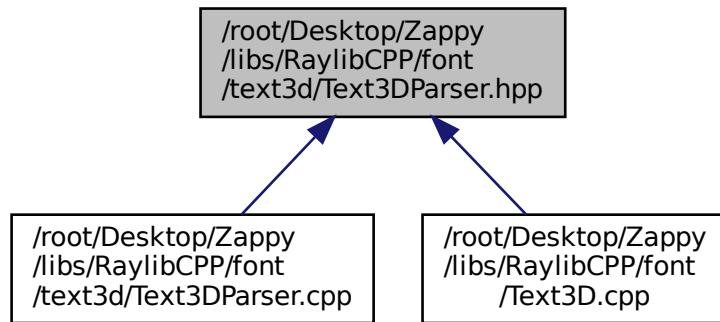
- `raylibcpp`

14.25 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DParser.hpp File Reference

```
#include <raylib.h>
#include <string>
#include <vector>
#include <memory>
#include "Text3DCore.hpp"
#include "Text3DHelper.hpp"
Include dependency graph for Text3DParser.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

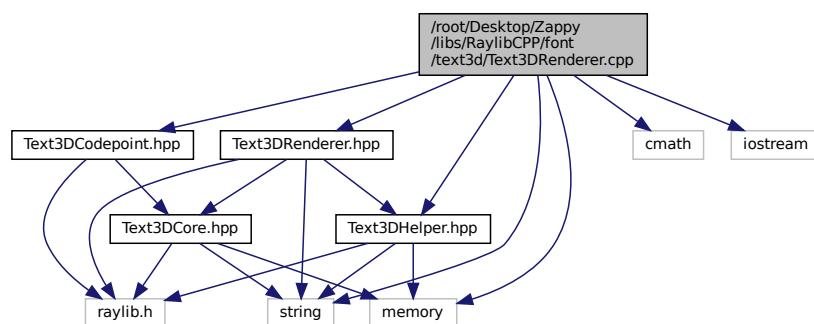
- class [raylibcpp::Text3DParser](#)

Namespaces

- [raylibcpp](#)

14.26 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DRenderer.cpp File Reference

```
#include "Text3DRenderer.hpp"
#include <cmath>
#include <iostream>
#include <string>
#include <memory>
#include "Text3DCodepoint.hpp"
#include "Text3DHelper.hpp"
Include dependency graph for Text3DRenderer.cpp:
```

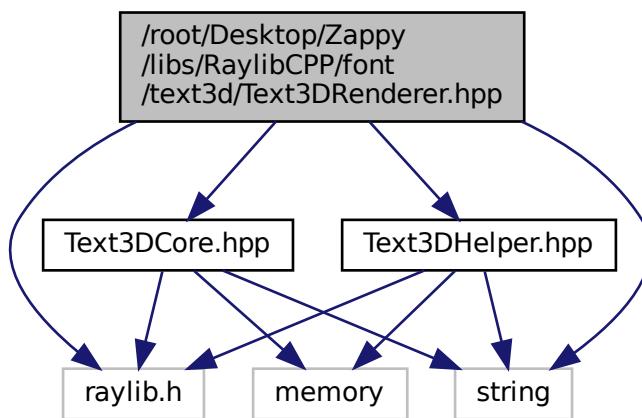


Namespaces

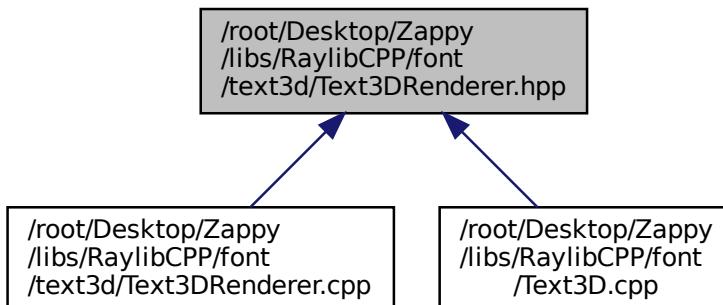
- [raylibcpp](#)

14.27 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DRenderer.hpp File Reference

```
#include <raylib.h>
#include <string>
#include "Text3DCore.hpp"
#include "Text3DHelper.hpp"
Include dependency graph for Text3DRenderer.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

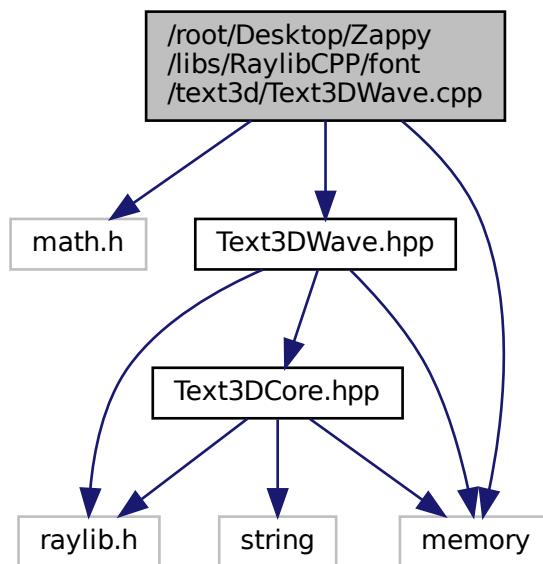
- class [raylibcpp::Text3DRenderer](#)

Namespaces

- raylibcpp

14.28 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DWave.cpp File Reference

```
#include <math.h>
#include <memory>
#include "Text3DWave.hpp"
Include dependency graph for Text3DWave.cpp:
```



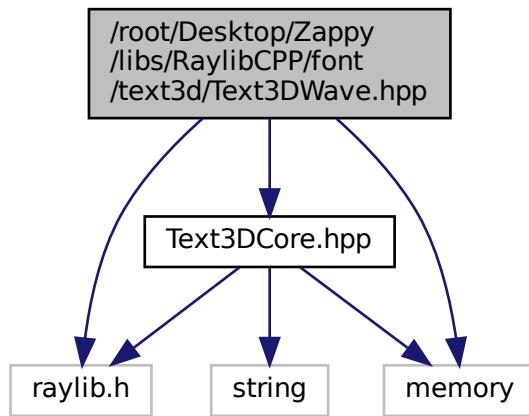
Namespaces

- raylibcpp

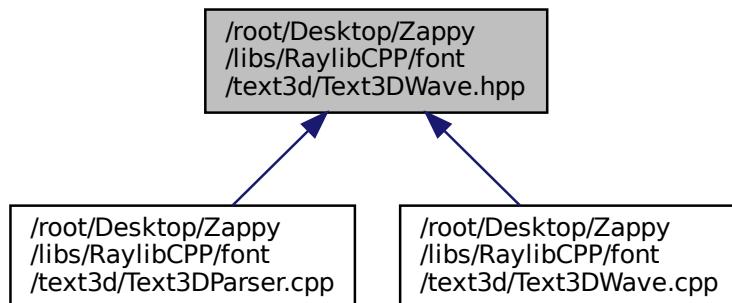
14.29 /root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DWave.hpp File Reference

```
#include <raylib.h>
#include <memory>
#include "Text3DCore.hpp"
```

Include dependency graph for Text3DWave.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [raylibcpp::Text3DWave](#)

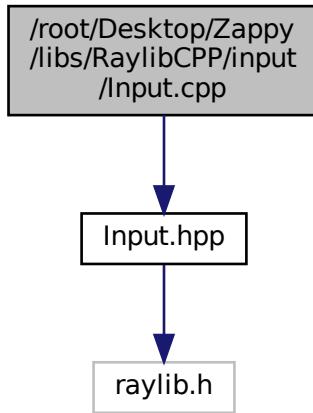
Namespaces

- [raylibcpp](#)

14.30 /root/Desktop/Zappy/libs/RaylibCPP/input/Input.cpp File Reference

```
#include "Input.hpp"
```

Include dependency graph for Input.cpp:



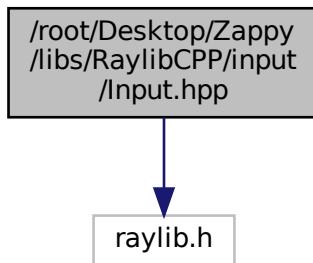
Namespaces

- [raylibcpp](#)

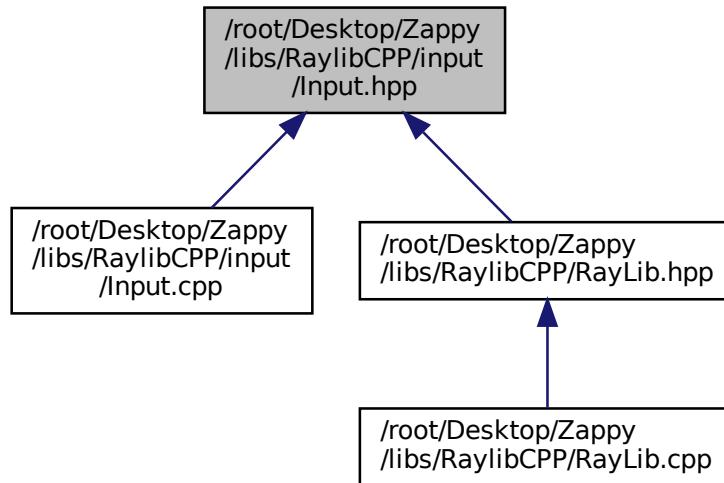
14.31 /root/Desktop/Zappy/libs/RaylibCPP/input/Input.hpp File Reference

```
#include <raylib.h>
```

Include dependency graph for Input.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [raylibcpp::Input](#)

Namespaces

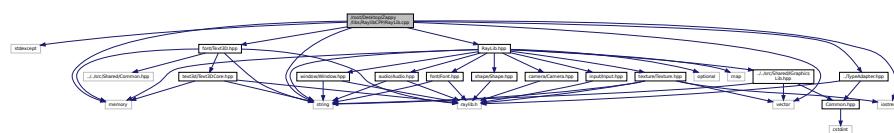
- [raylibcpp](#)

14.32 /root/Desktop/Zappy/libs/RaylibCPP/RayLib.cpp File Reference

```

#include <stdexcept>
#include <memory>
#include <string>
#include <vector>
#include <iostream>
#include "RayLib.hpp"
#include "../TypeAdapter.hpp"
#include "font/Text3D.hpp"
  
```

Include dependency graph for RayLib.cpp:



Functions

- `std::shared_ptr< IGraphicsLib > createGraphicsLib ()`

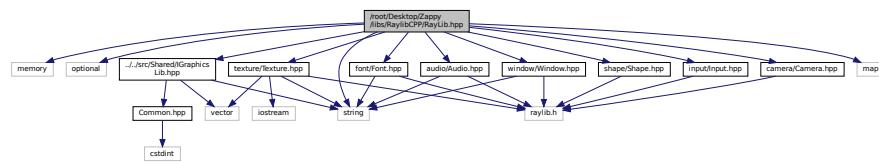
14.32.1 Function Documentation

14.32.1.1 createGraphicsLib()

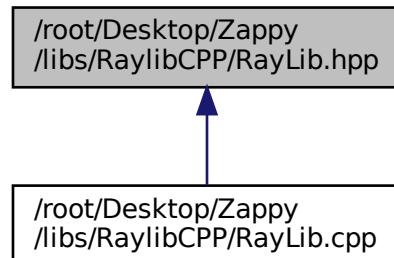
```
std::shared_ptr<IGraphicsLib> createGraphicsLib ( )
Definition at line 357 of file RayLib.cpp.
357
358     static std::shared_ptr<RayLib> instance = std::make_shared<RayLib>();
359     return instance;
360 }
```

14.33 /root/Desktop/Zappy/libs/RaylibCPP/RayLib.hpp File Reference

```
#include <memory>
#include <optional>
#include <string>
#include <map>
#include "../../src/Shared/IGraphicsLib.hpp"
#include "window/Window.hpp"
#include "shape/Shape.hpp"
#include "texture/Texture.hpp"
#include "font/Font.hpp"
#include "input/Input.hpp"
#include "audio/Audio.hpp"
#include "camera/Camera.hpp"
Include dependency graph for RayLib.hpp:
```



This graph shows which files directly or indirectly include this file:

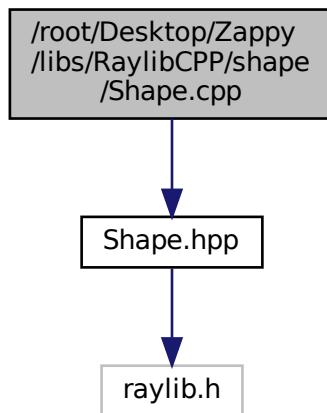


Data Structures

- class [RayLib](#)

14.34 /root/Desktop/Zappy/libs/RaylibCPP/shape/Shape.cpp File Reference

```
#include "Shape.hpp"  
Include dependency graph for Shape.cpp:
```

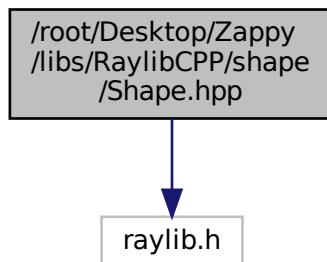


Namespaces

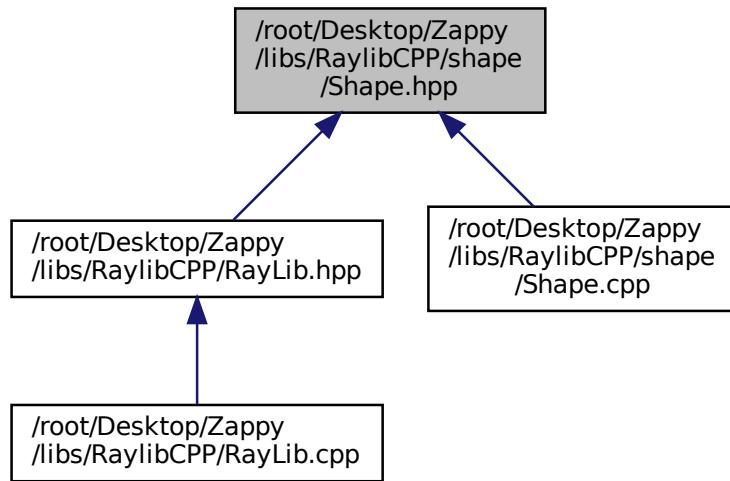
- [raylibcpp](#)

14.35 /root/Desktop/Zappy/libs/RaylibCPP/shape/Shape.hpp File Reference

```
#include <raylib.h>  
Include dependency graph for Shape.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [raylibcpp::Shape](#)

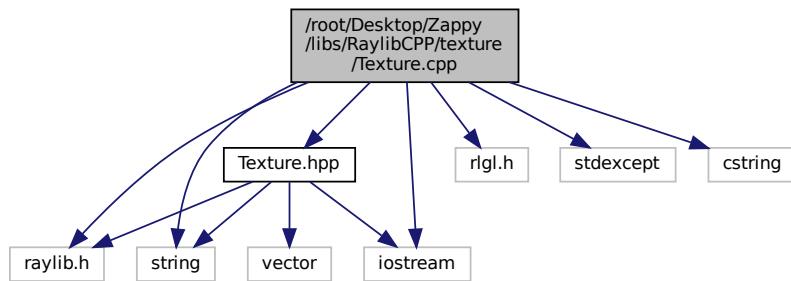
Namespaces

- [raylibcpp](#)

14.36 /root/Desktop/Zappy/libs/RaylibCPP/texture/Texture.cpp File Reference

```
#include "Texture.hpp"
#include <raylib.h>
#include <rlgl.h>
#include <stdexcept>
#include <iostream>
#include <cstring>
#include <string>
```

Include dependency graph for Texture.cpp:



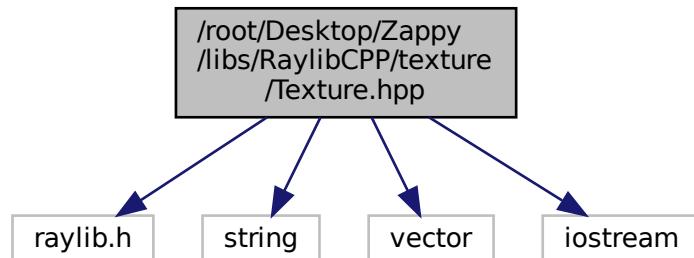
Namespaces

- [raylibcpp](#)

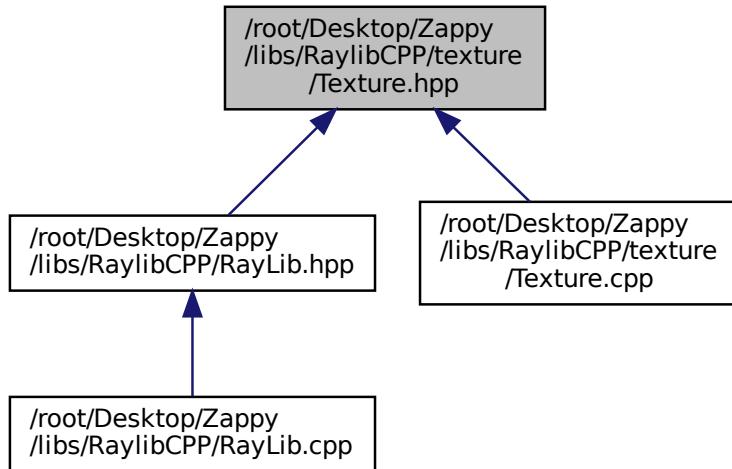
14.37 /root/Desktop/Zappy/libs/RaylibCPP/texture/Texture.hpp File Reference

```
#include <raylib.h>
#include <string>
#include <vector>
#include <iostream>
```

Include dependency graph for Texture.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class `raylibcpp::Texture`
- class `raylibcpp::Texture3DWrap`
- class `raylibcpp::ModelWrap`

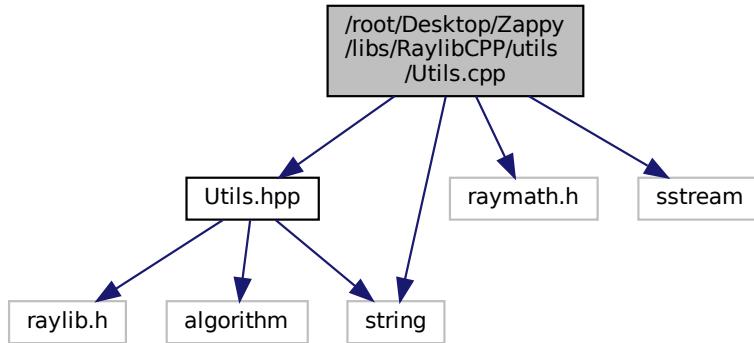
Namespaces

- `raylibcpp`

14.38 /root/Desktop/Zappy/libs/RaylibCPP/utils/Utils.cpp File Reference

```
#include "Utils.hpp"
#include <raymath.h>
#include <iostream>
#include <string>
```

Include dependency graph for Utils.cpp:

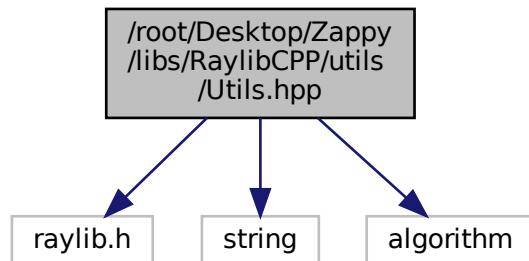


Namespaces

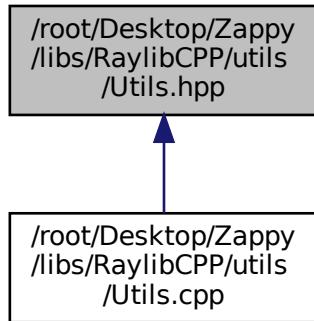
- [raylibcpp](#)

14.39 /root/Desktop/Zappy/libs/RaylibCPP/utils/Utils.hpp File Reference

```
#include <raylib.h>
#include <string>
#include <algorithm>
Include dependency graph for Utils.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [raylibcpp::Utils](#)
- class [raylibcpp::Vector3Wrap](#)

Namespaces

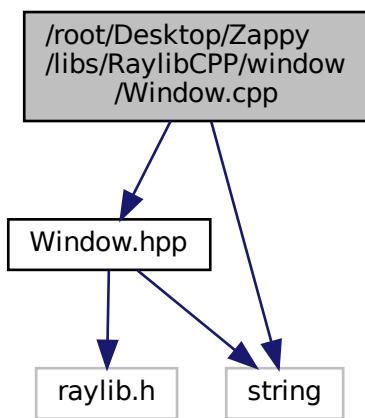
- [raylibcpp](#)

14.40 /root/Desktop/Zappy/libs/RaylibCPP/window/Window.cpp File Reference

```
#include "Window.hpp"
```

```
#include <string>
```

Include dependency graph for Window.cpp:



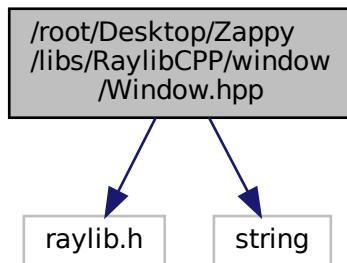
Namespaces

- [raylibcpp](#)

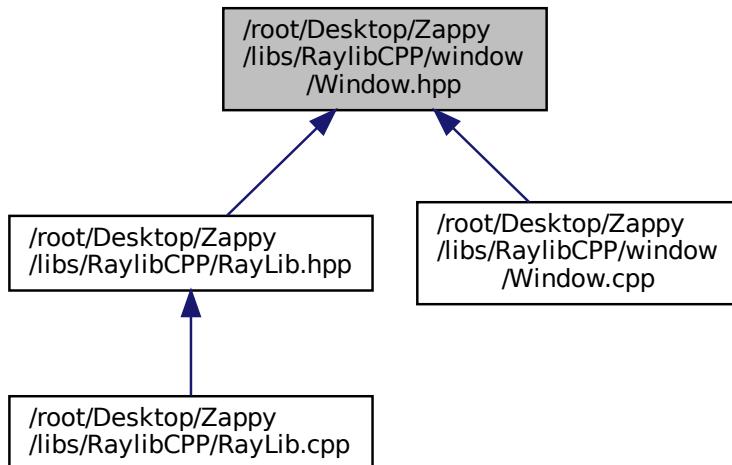
14.41 /root/Desktop/Zappy/libs/RaylibCPP/window/Window.hpp File Reference

```
#include <raylib.h>
#include <string>
```

Include dependency graph for Window.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [raylibcpp::Window](#)

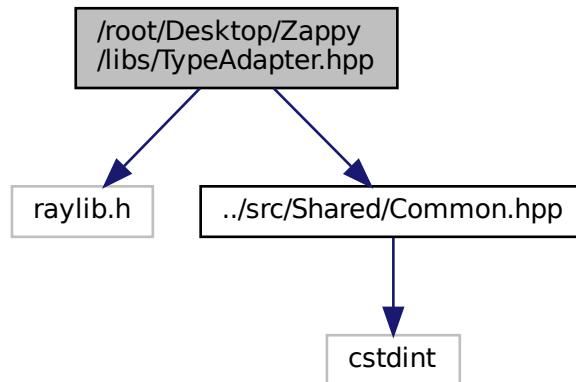
Namespaces

- [raylibcpp](#)

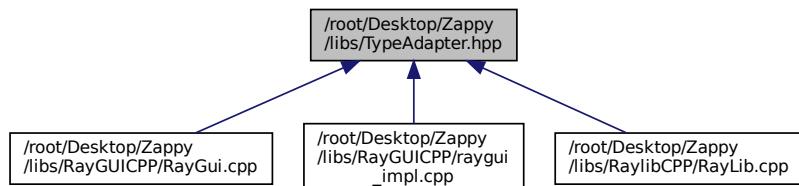
14.42 /root/Desktop/Zappy/libs/TypeAdapter.hpp File Reference

```
#include <raylib.h>
#include "../src/Shared/Common.hpp"
```

Include dependency graph for TypeAdapter.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- [TypeAdapter](#)

Functions

- `inline ::Vector2 TypeAdapter::ToRaylib (const ZappyTypes::Vector2 &vec)`
- `ZappyTypes::Vector2 TypeAdapter::FromRaylib (const ::Vector2 &vec)`
- `inline ::Vector3 TypeAdapter::ToRaylib (const ZappyTypes::Vector3 &vec)`
- `ZappyTypes::Vector3 TypeAdapter::FromRaylib (const ::Vector3 &vec)`
- `inline ::Color TypeAdapter::ToRaylib (const ZappyTypes::Color &color)`
- `ZappyTypes::Color TypeAdapter::FromRaylib (const ::Color &color)`
- `inline ::Rectangle TypeAdapter::ToRaylib (const ZappyTypes::Rectangle &rect)`
- `ZappyTypes::Rectangle TypeAdapter::FromRaylib (const ::Rectangle &rect)`

14.43 /root/Desktop/Zappy/src/AI/agent/actions.py File Reference

Namespaces

- [actions](#)

Functions

- def [actions.go_to_pos_with_distance](#) (agent, distance)
- def [actions.take_all_of_item_here](#) (agent, item)
- def [actions.take_everything_here](#) (agent)
- def [actions.go_take_item](#) (agent, item)
- def [actions.got_to_dir](#) (agent, direction)

14.44 /root/Desktop/Zappy/src/AI/agent/agent.py File Reference

Data Structures

- class [agent.Agent](#)

Namespaces

- [agent](#)

14.45 /root/Desktop/Zappy/src/AI/agent/behaviors.py File Reference

Data Structures

- class [behaviors.Behavior](#)
- class [behaviors.GetFoodBehavior](#)
- class [behaviors.UpgradeBehavior](#)
- class [behaviors.GetMineralsBehavior](#)
- class [behaviors.DysonBehavior](#)
- class [behaviors.GetFoodAndMineralsBehavior](#)

Namespaces

- [behaviors](#)

14.46 /root/Desktop/Zappy/src/AI/agent/broadcastManager.py File Reference

Data Structures

- class [broadcastManager.BroadcastManager](#)

Namespaces

- [broadcastManager](#)

14.47 /root/Desktop/Zappy/src/AI/agent/decisionManager.py File Reference

Data Structures

- class [decisionManager.DecisionManager](#)

Namespaces

- [decisionManager](#)

14.48 /root/Desktop/Zappy/src/AI/agent/socketManager.py File Reference

Data Structures

- class [socketManager.SocketManager](#)

Namespaces

- [socketManager](#)

14.49 /root/Desktop/Zappy/src/AI/constants/resources.py File Reference

Namespaces

- [resources](#)

Variables

- dictionary [resources.resources](#)

14.50 /root/Desktop/Zappy/src/AI/constants/roles.py File Reference

Namespaces

- [roles](#)

Variables

- dictionary [roles.roles](#)

14.51 /root/Desktop/Zappy/src/AI/constants/upgrades.py File Reference

Namespaces

- [upgrades](#)

14.52 /root/Desktop/Zappy/src/AI/logger/logger.py File Reference

Data Structures

- class [logger.LogLevel](#)
- class [logger.Logger](#)

Namespaces

- [logger](#)

14.53 /root/Desktop/Zappy/src/AI/main.py File Reference

Namespaces

- [main](#)

Functions

- def `main.run_agent` (ip, port, team, agent_id)
- def `main.signal_handler` (sig, frame)

Variables

- `main.args` = parseArgs()
- int `main.num_agents` = 1
- `main.port` = args.getPort()
- `main.ip` = args.getMachine()
- `main.team` = args.getName()
- list `main.child_pids` = []
- `main.pid` = os.fork()

14.54 /root/Desktop/Zappy/src/AI/parser/paringArgsClass.py File Reference

Data Structures

- class `paringArgsClass.ParsingArgsError`
- class `paringArgsClass.parseArgs`

Namespaces

- `paringArgsClass`

14.55 /root/Desktop/Zappy/src/AI/requirements.txt File Reference

14.56 /root/Desktop/Zappy/src/AI/utils/encryption.py File Reference

Namespaces

- `encryption`

Functions

- def `encryption.encrypt_message` (message)
- def `encryption.decrypt_message` (encrypted_message)

Variables

- `encryption.secret_key` = os.getenv("SECRET_KEY", "default_secret_key")

14.57 /root/Desktop/Zappy/src/AI/utils/input.py File Reference

Namespaces

- `input`

Functions

- def `input.get_key` ()

14.58 /root/Desktop/Zappy/src/AI/utils/zappy.py File Reference

Namespaces

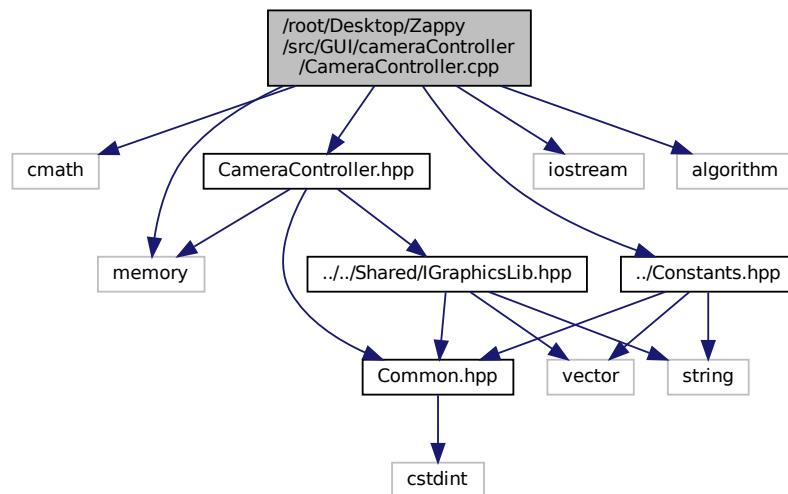
- zappy

Functions

- def `zappy.get_item_relative_pos` (item_position)
- def `zappy.parse_inventory` (inventory_str)
- def `zappy.get_closest_of_item` (surroundings_str, item)
- def `zappy.go_get_item` (surroundings, item)
- def `zappy.get_best_available_resource` (surroundings)
- def `zappy.how_much_of_item_here` (surroundings, item)

14.59 /root/Desktop/Zappy/src/GUI/cameraController/CameraController.cpp File Reference

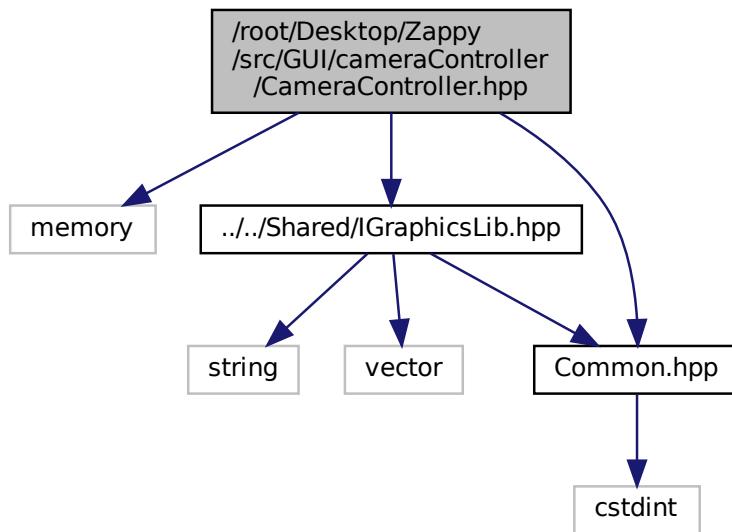
```
#include <cmath>
#include <memory>
#include <iostream>
#include <algorithm>
#include "CameraController.hpp"
#include "../Constants.hpp"
Include dependency graph for CameraController.cpp:
```



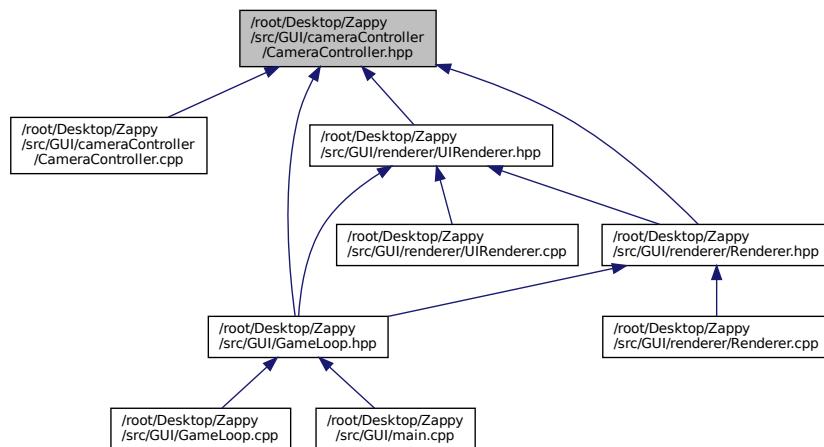
14.60 /root/Desktop/Zappy/src/GUI/cameraController/CameraController.hpp File Reference

```
#include <memory>
#include "../..../Shared/IGraphicsLib.hpp"
```

```
#include "../../Shared/Common.hpp"
Include dependency graph for CameraController.hpp:
```



This graph shows which files directly or indirectly include this file:



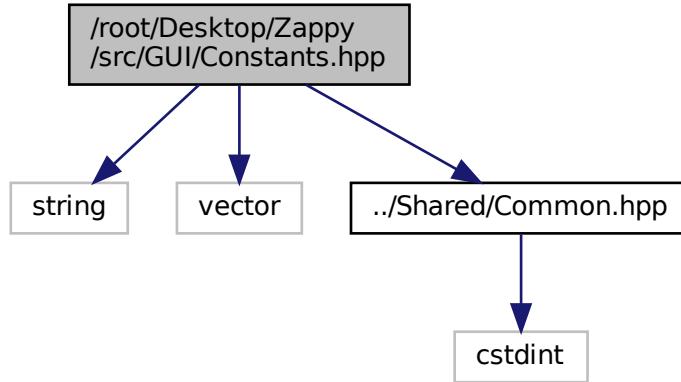
Data Structures

- class [CameraController](#)

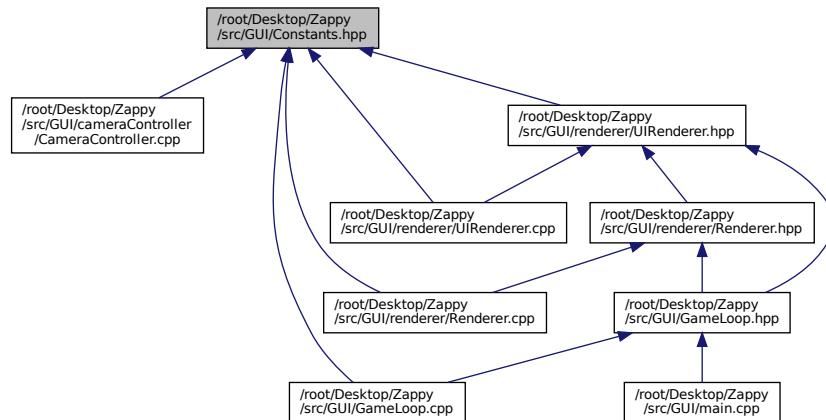
14.61 /root/Desktop/Zappy/src/GUI/Constants.hpp File Reference

```
#include <string>
#include <vector>
```

```
#include "../Shared/Common.hpp"
Include dependency graph for Constants.hpp:
```



This graph shows which files directly or indirectly include this file:



Enumerations

- enum `KeyboardKey` {
 `KEY_NULL = 0` , `KEY_A = 65` , `KEY_B = 66` , `KEY_C = 67` ,
 `KEY_D = 68` , `KEY_E = 69` , `KEY_F = 70` , `KEY_G = 71` ,
 `KEY_H = 72` , `KEY_I = 73` , `KEY_J = 74` , `KEY_K = 75` ,
 `KEY_L = 76` , `KEY_M = 77` , `KEY_N = 78` , `KEY_O = 79` ,
 `KEY_P = 80` , `KEY_Q = 81` , `KEY_R = 82` , `KEY_S = 83` ,
 `KEY_T = 84` , `KEY_U = 85` , `KEY_V = 86` , `KEY_W = 87` ,
 `KEY_X = 88` , `KEY_Y = 89` , `KEY_Z = 90` , `KEY_F1 = 290` ,
 `KEY_F2 = 291` , `KEY_F3 = 292` , `KEY_RIGHT = 262` , `KEY_LEFT = 263` ,
 `KEY_DOWN = 264` , `KEY_UP = 265` , `KEY_SPACE = 32` , `KEY_ESCAPE = 256` ,
 `KEY_ENTER = 257` , `KEY_TAB = 258` , `KEY_BACKSPACE = 259` , `KEY_INSERT = 260` ,
 `KEY_DELETE = 261` , `KEY_SHIFT_LEFT = 340` , `KEY_CONTROL_LEFT = 341` , `KEY_ALT_LEFT = 342` ,
 `KEY_SUPER_LEFT = 343` }

Variables

- const int `DEFAULT_WIDTH` = 1280
- const int `DEFAULT_HEIGHT` = 720
- const char `DEFAULT_TITLE` [] = "Zappy GUI"
- const int `NUM_RANDOM_MODELS` = 20

14.61.1 Enumeration Type Documentation

14.61.1.1 KeyboardKey

enum `KeyboardKey`

Enumerator

KEY_NULL	
KEY_A	
KEY_B	
KEY_C	
KEY_D	
KEY_E	
KEY_F	
KEY_G	
KEY_H	
KEY_I	
KEY_J	
KEY_K	
KEY_L	
KEY_M	
KEY_N	
KEY_O	
KEY_P	
KEY_Q	
KEY_R	
KEY_S	
KEY_T	
KEY_U	
KEY_V	
KEY_W	
KEY_X	
KEY_Y	
KEY_Z	
KEY_F1	
KEY_F2	
KEY_F3	
KEY_RIGHT	
KEY_LEFT	
KEY_DOWN	
KEY_UP	
KEY_SPACE	
KEY_ESCAPE	
KEY_ENTER	

Enumerator

KEY_TAB	
KEY_BACKSPACE	
KEY_INSERT	
KEY_DELETE	
KEY_SHIFT_LEFT	
KEY_CONTROL_LEFT	
KEY_ALT_LEFT	
KEY_SUPER_LEFT	

Definition at line 19 of file Constants.hpp.

```

19      {
20      KEY_NULL          = 0,
21      KEY_A            = 65,
22      KEY_B            = 66,
23      KEY_C            = 67,
24      KEY_D            = 68,
25      KEY_E            = 69,
26      KEY_F            = 70,
27      KEY_G            = 71,
28      KEY_H            = 72,
29      KEY_I            = 73,
30      KEY_J            = 74,
31      KEY_K            = 75,
32      KEY_L            = 76,
33      KEY_M            = 77,
34      KEY_N            = 78,
35      KEY_O            = 79,
36      KEY_P            = 80,
37      KEY_Q            = 81,
38      KEY_R            = 82,
39      KEY_S            = 83,
40      KEY_T            = 84,
41      KEY_U            = 85,
42      KEY_V            = 86,
43      KEY_W            = 87,
44      KEY_X            = 88,
45      KEY_Y            = 89,
46      KEY_Z            = 90,
47
48      KEY_F1           = 290,
49      KEY_F2           = 291,
50      KEY_F3           = 292,
51
52      KEY_RIGHT         = 262,
53      KEY_LEFT          = 263,
54      KEY_DOWN          = 264,
55      KEY_UP           = 265,
56
57      KEY_SPACE          = 32,
58      KEY_ESCAPE         = 256,
59      KEY_ENTER          = 257,
60      KEY_TAB            = 258,
61      KEY_BACKSPACE      = 259,
62      KEY_INSERT         = 260,
63      KEY_DELETE         = 261,
64
65      KEY_SHIFT_LEFT     = 340,
66      KEY_CONTROL_LEFT   = 341,
67      KEY_ALT_LEFT       = 342,
68      KEY_SUPER_LEFT     = 343
69  };

```

14.61.2 Variable Documentation

14.61.2.1 DEFAULT_HEIGHT

```
const int DEFAULT_HEIGHT = 720
```

Definition at line 15 of file Constants.hpp.

14.61.2.2 DEFAULT_TITLE

```
const char DEFAULT_TITLE[ ] = "Zappy GUI"
Definition at line 16 of file Constants.hpp.
```

14.61.2.3 DEFAULT_WIDTH

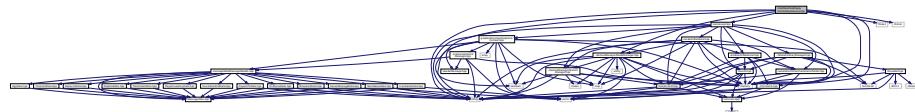
```
const int DEFAULT_WIDTH = 1280
Definition at line 14 of file Constants.hpp.
```

14.61.2.4 NUM_RANDOM_MODELS

```
const int NUM_RANDOM_MODELS = 20
Definition at line 17 of file Constants.hpp.
```

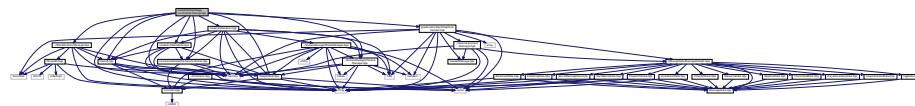
14.62 /root/Desktop/Zappy/src/GUI/GameLoop.cpp File Reference

```
#include "GameLoop.hpp"
#include <iostream>
#include <thread>
#include <memory>
#include <string>
#include <chrono>
#include "Constants.hpp"
#include "textureManager/ModelManager.hpp"
Include dependency graph for GameLoop.cpp:
```

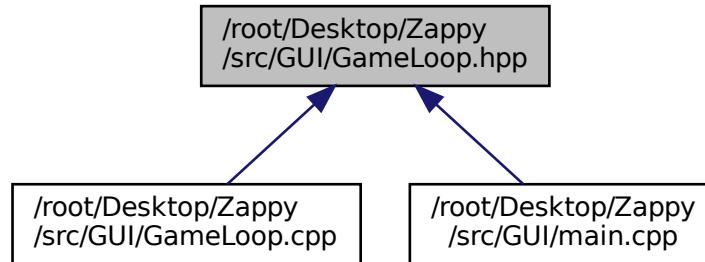


14.63 /root/Desktop/Zappy/src/GUI/GameLoop.hpp File Reference

```
#include <memory>
#include <string>
#include "../Shared/LibraryManager.hpp"
#include "../Shared/IGraphicsLib.hpp"
#include "../Shared/IGuiLib.hpp"
#include "cameraController/CameraController.hpp"
#include "renderer/UIRenderer.hpp"
#include "renderer/Renderer.hpp"
#include "textureManager/TextureManager.hpp"
#include "graphicalContext/GraphicalContext.hpp"
Include dependency graph for GameLoop.hpp:
```



This graph shows which files directly or indirectly include this file:

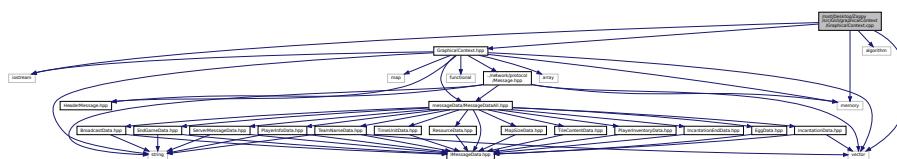


Data Structures

- class [GameLoop](#)

14.64 /root/Desktop/Zappy/src/GUI/graphicalContext/GraphicalContext.cpp File Reference

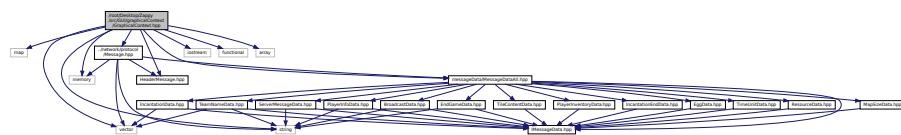
```
#include <iostream>
#include <vector>
#include <memory>
#include <algorithm>
#include "GraphicalContext.hpp"
Include dependency graph for GraphicalContext.cpp:
```



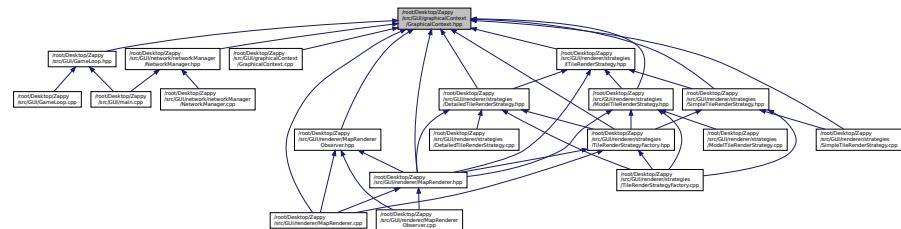
14.65 /root/Desktop/Zappy/src/GUI/graphicalContext/GraphicalContext.hpp File Reference

```
#include <map>
#include <vector>
#include <memory>
#include <iostream>
#include <functional>
#include <string>
#include <array>
#include "../network/protocol/Message.hpp"
#include "../network/protocol/messageData/MessageDataAll.hpp"
#include "../network/protocol/HeaderMessage.hpp"
```

Include dependency graph for GraphicalContext.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [TileData](#)
Structure représentant une tuile dans le contexte graphique.
 - class [IGraphicalContextObserver](#)
Observateur de changements de contexte graphique.
 - class [GraphicalContext](#)

Enumerations

- enum class ResourceType {
FOOD = 0 , LINEMATE , DERAUMERE , SIBUR ,
MENDIANE , PHIRAS , THYSTAME , COUNT }

Énumération des types de ressources.

14.65.1 Enumeration Type Documentation

14.65.1.1 ResourceType

enum ResourceType [strong]
Énumération des types de ressources.

Enumerator

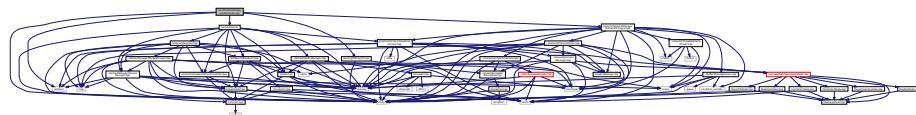
FOOD
LINEMATE
DERAUMERE
SIBUR
MENDIANE
PHIRAS
THYSTAME
COUNT

Definition at line 46 of file GraphicalContext.hpp.

```
46     {
47     FOOD = 0,
48     LINEMATE,
49     DERAUMERE,
50     SIBUR,
51     MENDIANE,
52     PHIRAS,
53     THYSTAME,
54     COUNT
55 };
```

14.66 /root/Desktop/Zappy/src/GUI/main.cpp File Reference

```
#include <iostream>
#include <string>
#include "parsing/ParsingCLI.hpp"
#include "GameLoop.hpp"
#include "network/networkManager/NetworkManager.hpp"
Include dependency graph for main.cpp:
```



Functions

- void [displayHelp\(\)](#)
- int [main](#) (int argc, char **argv)

14.66.1 Function Documentation

14.66.1.1 [displayHelp\(\)](#)

```
void displayHelp ( )
```

Definition at line 14 of file main.cpp.

```
14     {
15     std::cout << "USAGE: ./zappy_gui -p port -h machine" << std::endl;
16     std::cout << "\tport\this the port number" << std::endl;
17     std::cout << "\tmachine\this the name of the machine; localhost by default" << std::endl;
18 }
```

14.66.1.2 [main\(\)](#)

```
int main (
    int argc,
    char ** argv )
```

Definition at line 20 of file main.cpp.

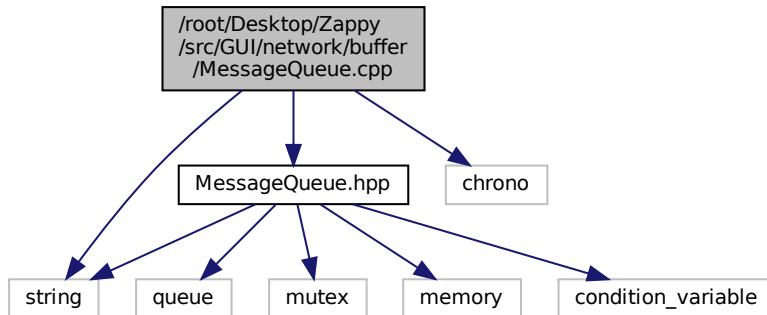
```
20     {
21     try {
22         if (argc == 2 && std::string(argv[1]) == "--help") {
23             displayHelp();
24             return 0;
25         }
26         ParsingCLI parser(argc, argv);
27         std::cout << "Connecting to " << parser.getMachine() << " on port " << parser.getPort() << std::endl;
28
29         NetworkManager networkManager;
30
31         if (!networkManager.connect(parser.getMachine(), parser.getPort())) {
32             std::cerr << "[ERROR] Impossible de se connecter au serveur." << std::endl;
33             return 84;
34     }
```

```

35     // pas sur que ca sois la bonne facon de faire mais le networkManager doit etre dans un thread
36     std::thread networkThread(&NetworkManager::networkThreadLoop, &networkManager);
37
38     GameLoop gameLoop;
39     gameLoop.setServerInfo(parser.getMachine(), parser.getPort());
40     if (!gameLoop.init()) {
41         std::cerr << "Failed to initialize game components" << std::endl;
42         return 84;
43     }
44     return gameLoop.run();
45 } catch (const AEException &e) {
46     std::cerr << e.getFormattedMessage() << std::endl;
47     return 84;
48 } catch (const std::exception &e) {
49     std::cerr << "Exception: " << e.what() << std::endl;
50     return 84;
51 } catch (...) {
52     std::cerr << "An unexpected error occurred." << std::endl;
53     return 84;
54 }
55 return 0;
56 }
```

14.67 /root/Desktop/Zappy/src/GUI/network/buffer/MessageQueue.cpp File Reference

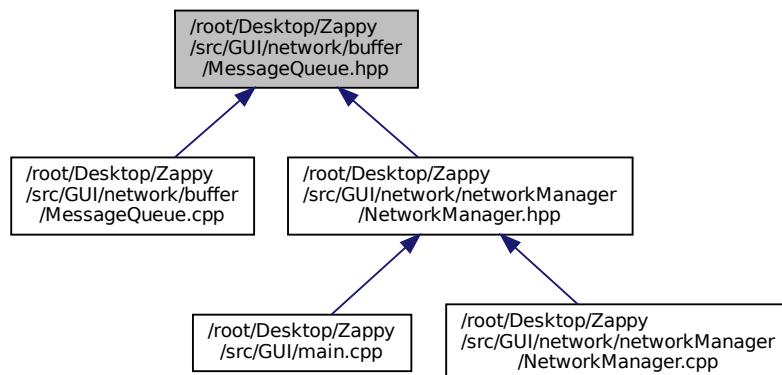
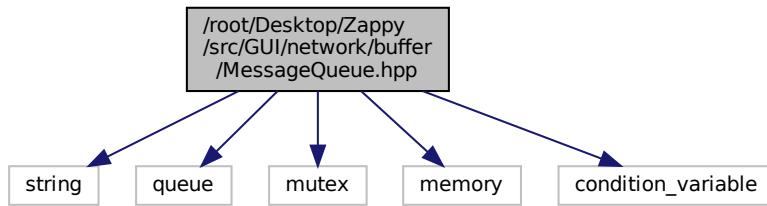
```
#include "MessageQueue.hpp"
#include <string>
#include <chrono>
Include dependency graph for MessageQueue.cpp:
```



14.68 /root/Desktop/Zappy/src/GUI/network/buffer/MessageQueue.hpp File Reference

```
#include <string>
#include <queue>
#include <mutex>
#include <memory>
#include <condition_variable>
```

Include dependency graph for MessageQueue.hpp:



Data Structures

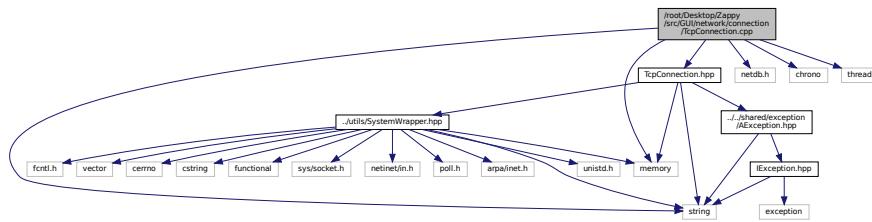
- class [MessageQueue](#)

14.69 /root/Desktop/Zappy/src/GUI/network/connection/TcpConnection.cpp File Reference

```

#include "TcpConnection.hpp"
#include <netdb.h>
#include <memory>
#include <string>
#include <chrono>
#include <thread>
  
```

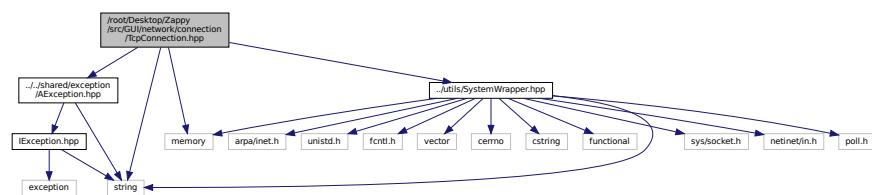
Include dependency graph for TcpConnection.cpp:



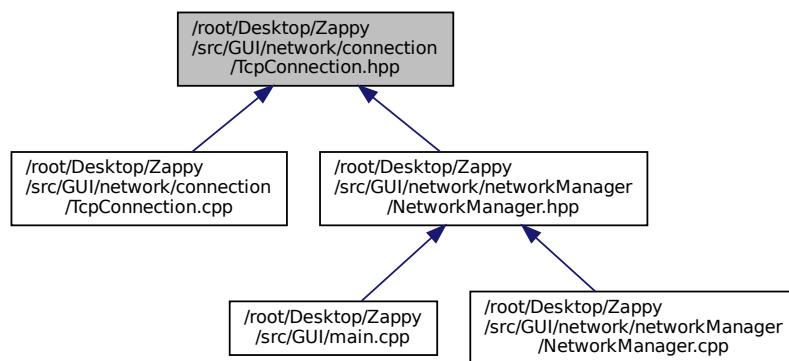
14.70 /root/Desktop/Zappy/src/GUI/network/connection/TcpConnection.hpp File Reference

```
#include <string>
#include <memory>
#include "../..../shared/exception/AException.hpp"
#include "../utils/SystemWrapper.hpp"
```

Include dependency graph for TcpConnection.hpp:



This graph shows which files directly or indirectly include this file:

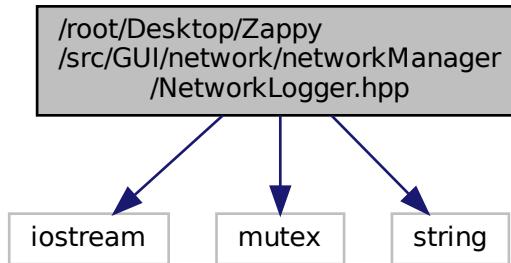


Data Structures

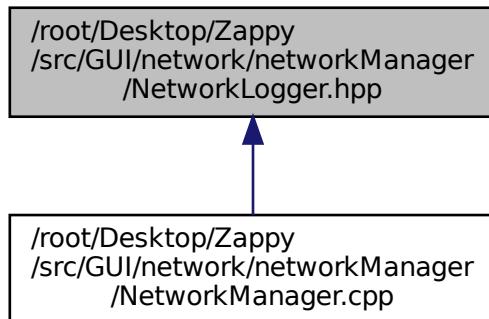
- class [TcpConnection](#)

14.71 /root/Desktop/Zappy/src/GUI/network/networkManager/NetworkLogger.hpp File Reference

```
#include <iostream>
#include <mutex>
#include <string>
Include dependency graph for NetworkLogger.hpp:
```



This graph shows which files directly or indirectly include this file:



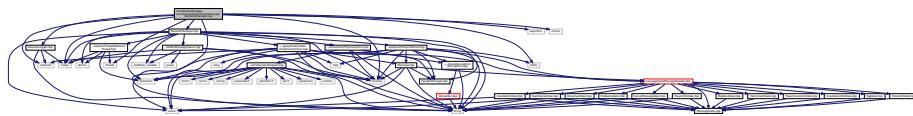
Data Structures

- class [NetworkLogger](#)

14.72 /root/Desktop/Zappy/src/GUI/network/networkManager/NetworkManager.cpp File Reference

```
#include <iostream>
#include <algorithm>
#include <thread>
#include <chrono>
```

```
#include <mutex>
#include <utility>
#include <string>
#include <memory>
#include "NetworkManager.hpp"
#include "NetworkLogger.hpp"
Include dependency graph for NetworkManager.cpp:
```

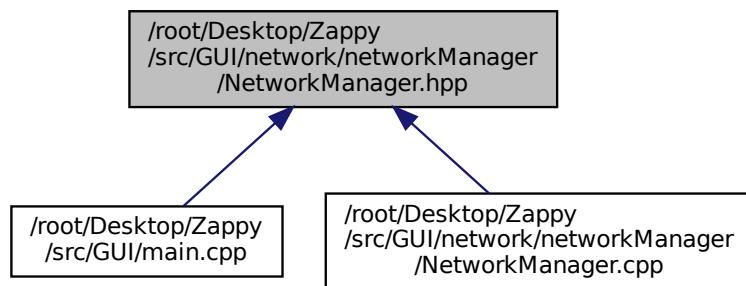


14.73 /root/Desktop/Zappy/src/GUI/network/networkManager/NetworkManager.hpp File Reference

```
#include <string>
#include <vector>
#include <functional>
#include <memory>
#include <mutex>
#include <atomic>
#include <condition_variable>
#include "../connection/TcpConnection.hpp"
#include "../protocol/ProtocolParser.hpp"
#include "../protocol/HeaderMessage.hpp"
#include "../networkThread/NetworkThread.hpp"
#include "../buffer/MessageQueue.hpp"
#include "../../graphicalContext/GraphicalContext.hpp"
Include dependency graph for NetworkManager.hpp:
```



This graph shows which files directly or indirectly include this file:

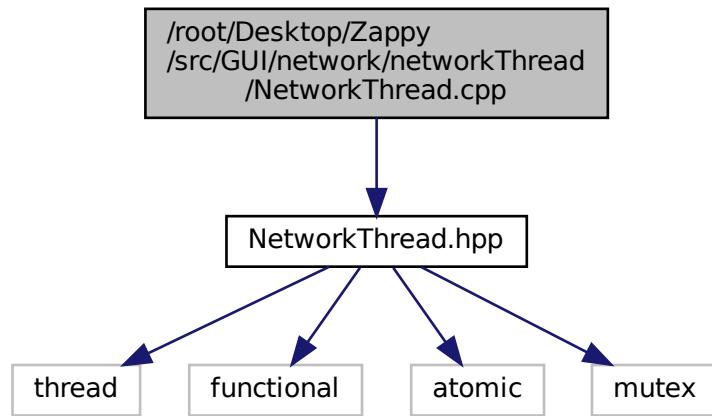


Data Structures

- class [NetworkManager](#)

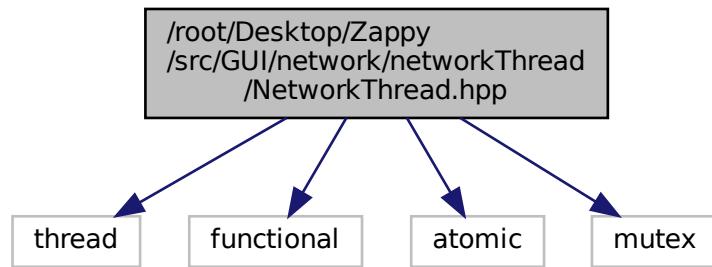
14.74 /root/Desktop/Zappy/src/GUI/network/networkThread/NetworkThread.cpp File Reference

```
#include "NetworkThread.hpp"  
Include dependency graph for NetworkThread.cpp:
```

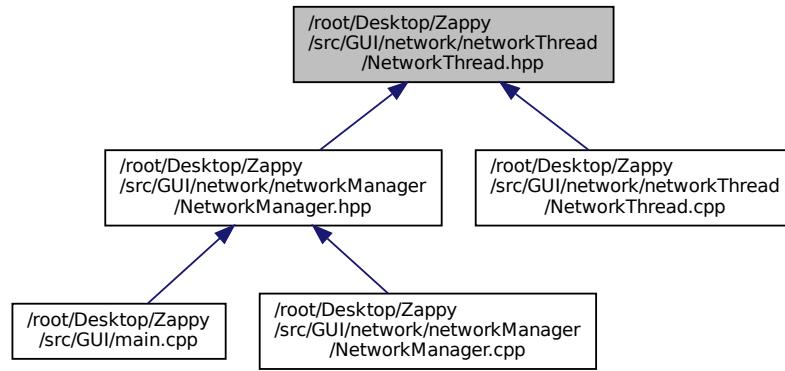


14.75 /root/Desktop/Zappy/src/GUI/network/networkThread/NetworkThread.hpp File Reference

```
#include <thread>  
#include <functional>  
#include <atomic>  
#include <mutex>  
Include dependency graph for NetworkThread.hpp:
```



This graph shows which files directly or indirectly include this file:

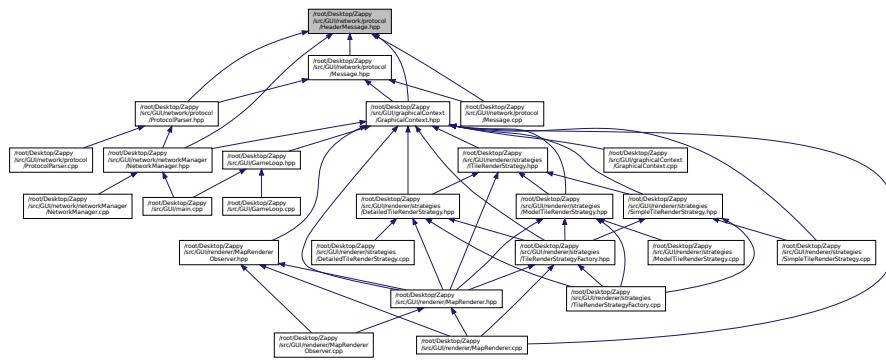


Data Structures

- class [NetworkThread](#)

14.76 /root/Desktop/Zappy/src/GUI/network/protocol/HeaderMessage.hpp File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define MSZ_HEADER "msz"
- #define BCT_HEADER "bct"
- #define MCT_HEADER "mct"
- #define TNA_HEADER "tna"
- #define PNW_HEADER "pnw"
- #define PPO_HEADER "ppo"
- #define PLV_HEADER "plv"
- #define PIN_HEADER "pin"
- #define PEX_HEADER "pex"
- #define PBC_HEADER "pbc"
- #define PDR_HEADER "pdr"

- #define PGT_HEADER "pgt"
- #define PDI_HEADER "pdi"
- #define PIC_HEADER "pic"
- #define PIE_HEADER "pie"
- #define ENW_HEADER "enw"
- #define EBO_HEADER "ebo"
- #define EDI_HEADER "edi"
- #define PFK_HEADER "pfk"
- #define SGT_HEADER "sgt"
- #define SST_HEADER "sst"
- #define SEG_HEADER "seg"
- #define SMG_HEADER "smg"
- #define SUC_HEADER "suc"
- #define SBP_HEADER "sbp"

14.76.1 Macro Definition Documentation

14.76.1.1 BCT_HEADER

```
#define BCT_HEADER "bct"
Definition at line 13 of file HeaderMessage.hpp.
```

14.76.1.2 EBO_HEADER

```
#define EBO_HEADER "ebo"
Definition at line 37 of file HeaderMessage.hpp.
```

14.76.1.3 EDI_HEADER

```
#define EDI_HEADER "edi"
Definition at line 38 of file HeaderMessage.hpp.
```

14.76.1.4 ENW_HEADER

```
#define ENW_HEADER "enw"
Definition at line 36 of file HeaderMessage.hpp.
```

14.76.1.5 MCT_HEADER

```
#define MCT_HEADER "mct"
Definition at line 14 of file HeaderMessage.hpp.
```

14.76.1.6 MSZ_HEADER

```
#define MSZ_HEADER "msz"
Definition at line 12 of file HeaderMessage.hpp.
```

14.76.1.7 PBC_HEADER

```
#define PBC_HEADER "pbc"
```

Definition at line 25 of file HeaderMessage.hpp.

14.76.1.8 PDI_HEADER

```
#define PDI_HEADER "pdi"
```

Definition at line 29 of file HeaderMessage.hpp.

14.76.1.9 PDR_HEADER

```
#define PDR_HEADER "pdr"
```

Definition at line 27 of file HeaderMessage.hpp.

14.76.1.10 PEX_HEADER

```
#define PEX_HEADER "pex"
```

Definition at line 24 of file HeaderMessage.hpp.

14.76.1.11 PFK_HEADER

```
#define PFK_HEADER "pfk"
```

Definition at line 39 of file HeaderMessage.hpp.

14.76.1.12 PGT_HEADER

```
#define PGT_HEADER "pgt"
```

Definition at line 28 of file HeaderMessage.hpp.

14.76.1.13 PIC_HEADER

```
#define PIC_HEADER "pic"
```

Definition at line 32 of file HeaderMessage.hpp.

14.76.1.14 PIE_HEADER

```
#define PIE_HEADER "pie"
```

Definition at line 33 of file HeaderMessage.hpp.

14.76.1.15 PIN_HEADER

```
#define PIN_HEADER "pin"
```

Definition at line 23 of file HeaderMessage.hpp.

14.76.1.16 PLV_HEADER

```
#define PLV_HEADER "plv"
```

Definition at line 22 of file HeaderMessage.hpp.

14.76.1.17 PNW_HEADER

```
#define PNW_HEADER "pnw"
Definition at line 20 of file HeaderMessage.hpp.
```

14.76.1.18 PPO_HEADER

```
#define PPO_HEADER "ppo"
Definition at line 21 of file HeaderMessage.hpp.
```

14.76.1.19 SBP_HEADER

```
#define SBP_HEADER "sbp"
Definition at line 51 of file HeaderMessage.hpp.
```

14.76.1.20 SEG_HEADER

```
#define SEG_HEADER "seg"
Definition at line 46 of file HeaderMessage.hpp.
```

14.76.1.21 SGT_HEADER

```
#define SGT_HEADER "sgt"
Definition at line 42 of file HeaderMessage.hpp.
```

14.76.1.22 SMG_HEADER

```
#define SMG_HEADER "smg"
Definition at line 47 of file HeaderMessage.hpp.
```

14.76.1.23 SST_HEADER

```
#define SST_HEADER "sst"
Definition at line 43 of file HeaderMessage.hpp.
```

14.76.1.24 SUC_HEADER

```
#define SUC_HEADER "suc"
Definition at line 50 of file HeaderMessage.hpp.
```

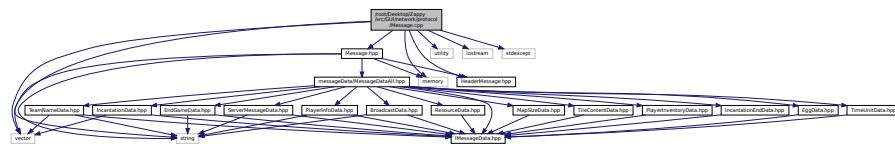
14.76.1.25 TNA_HEADER

```
#define TNA_HEADER "tna"
Definition at line 17 of file HeaderMessage.hpp.
```

14.77 /root/Desktop/Zappy/src/GUI/network/protocol/Message.cpp File Reference

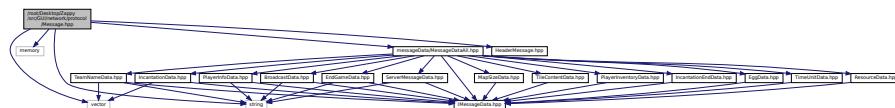
```
#include "Message.hpp"
#include <utility>
#include <iostream>
```

```
#include <string>
#include <stdexcept>
#include <vector>
#include <memory>
#include "HeaderMessage.hpp"
Include dependency graph for Message.cpp:
```

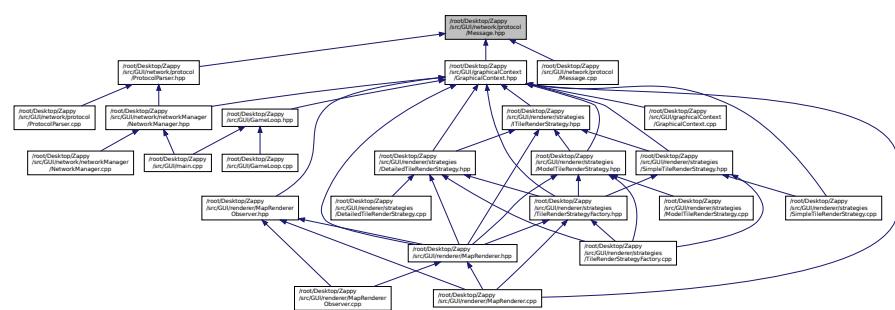


14.78 /root/Desktop/Zappy/src/GUI/network/protocol/Message.hpp File Reference

```
#include <string>
#include <memory>
#include <vector>
#include "messageData/MessageDataAll.hpp"
#include "HeaderMessage.hpp"
Include dependency graph for Message.hpp:
```



This graph shows which files directly or indirectly include this file:



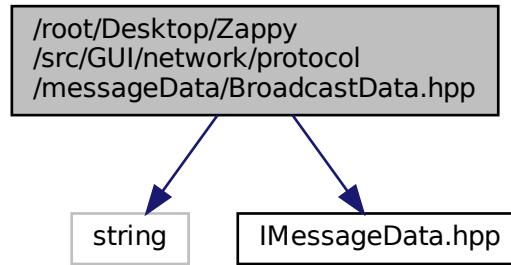
Data Structures

- class [Message](#)

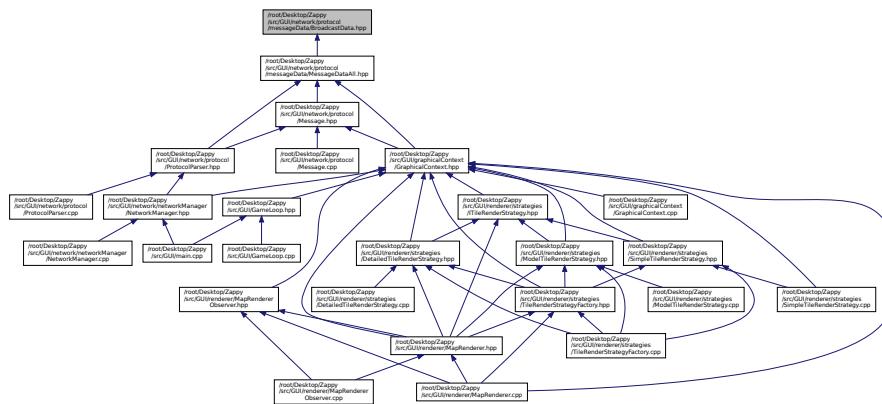
14.79 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/BroadcastData.hpp File Reference

```
#include <string>
#include "IMessageData.hpp"
```

Include dependency graph for BroadcastData.hpp:



This graph shows which files directly or indirectly include this file:



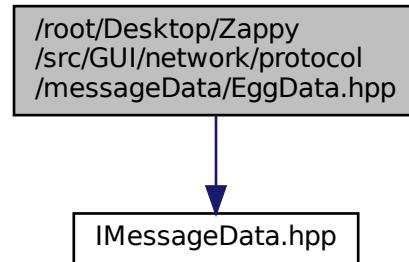
Data Structures

- class [BroadcastData](#)

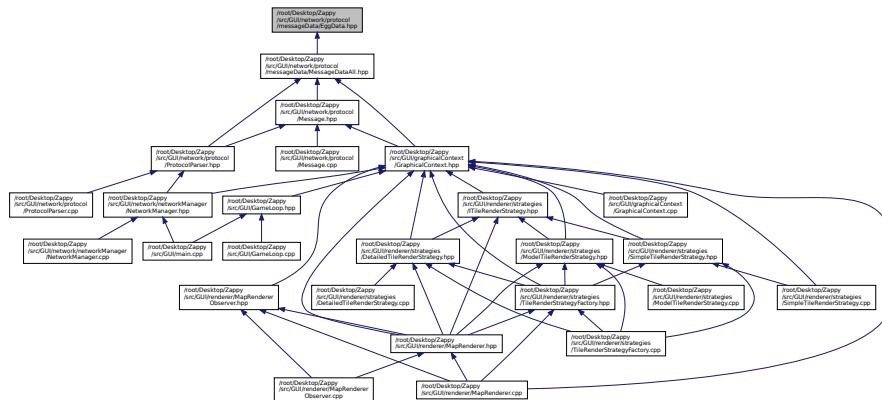
14.80 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/EggData.hpp File Reference

```
#include "IMessageData.hpp"
```

Include dependency graph for EggData.hpp:



This graph shows which files directly or indirectly include this file:



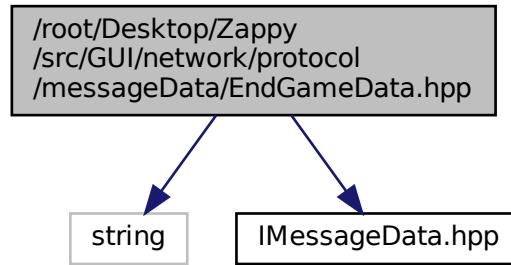
Data Structures

- class [EggData](#)

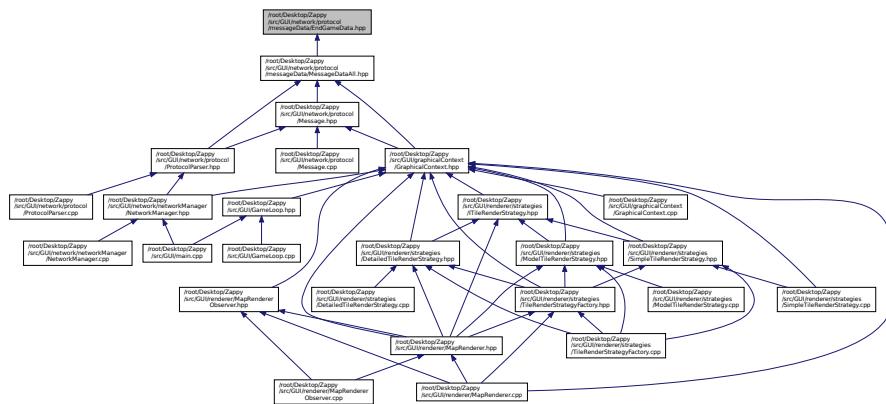
14.81 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/End ← GameData.hpp File Reference

```
#include <string>
#include "IMessageData.hpp"
```

Include dependency graph for EndGameData.hpp:



This graph shows which files directly or indirectly include this file:

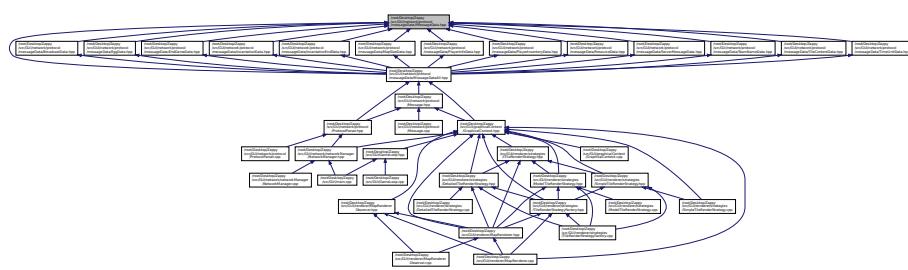


Data Structures

- class [EndGameData](#)

14.82 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/IMessageData.hpp File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- class [IMessageData](#)

Enumerations

- enum class [MessageType](#) {

MapSize , TileContent , PlayerInfo , PlayerInventory ,

Broadcast , Incantation , IncantationEnd , Egg ,

TimeUnit , EndGame , ServerMessage , TeamName ,

Resource }

14.82.1 Enumeration Type Documentation

14.82.1.1 MessageType

enum [MessageType](#) [strong]

Enumerator

MapSize	
TileContent	
PlayerInfo	
PlayerInventory	
Broadcast	
Incantation	
IncantationEnd	
Egg	
TimeUnit	
EndGame	
ServerMessage	
TeamName	
Resource	

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

```

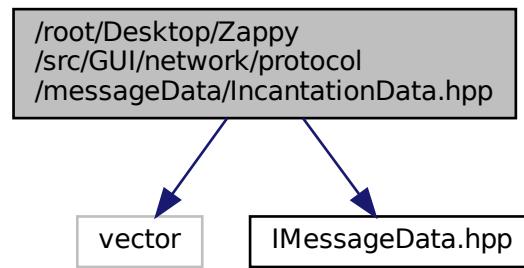
11
12     MapSize,
13     TileContent,
14     PlayerInfo,
15     PlayerInventory,
16     Broadcast,
17     Incantation,
18     IncantationEnd,
19     Egg,
20     TimeUnit,
21     EndGame,
22     ServerMessage,
23     TeamName,
24     Resource,
25 };

```

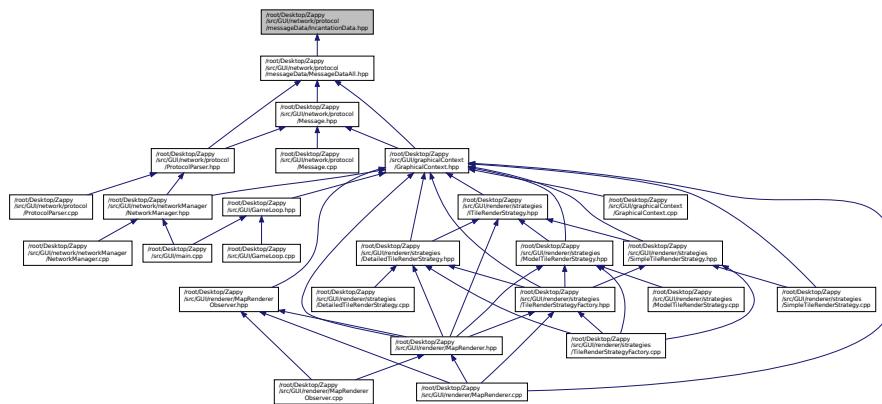
14.83 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/[IncantationData.hpp](#) File Reference

```
#include <vector>
#include "IMessageData.hpp"
```

Include dependency graph for IncantationData.hpp:



This graph shows which files directly or indirectly include this file:



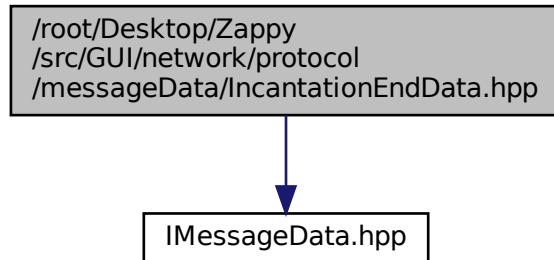
Data Structures

- class [IncantationData](#)

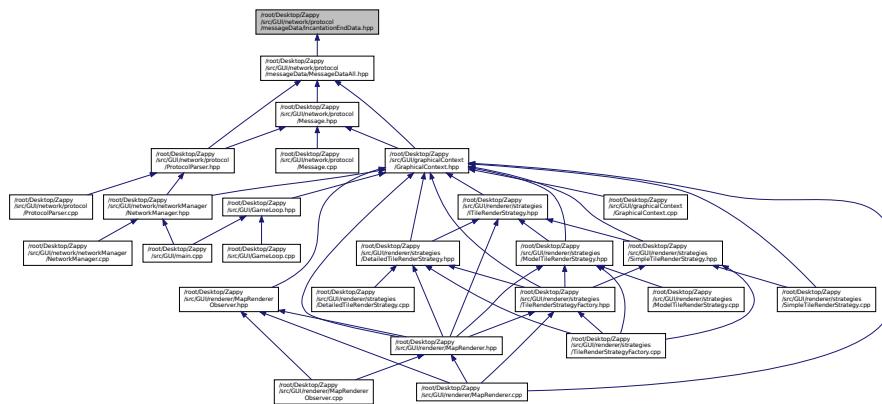
14.84 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/IncantationEndData.hpp File Reference

```
#include "IMessageData.hpp"
```

Include dependency graph for IncantationEndData.hpp:



This graph shows which files directly or indirectly include this file:



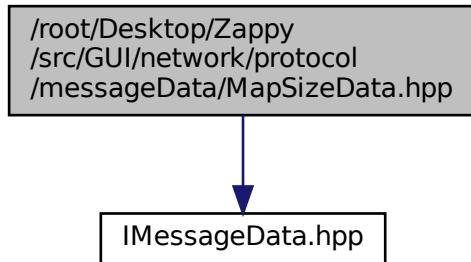
Data Structures

- class [IncantationEndData](#)

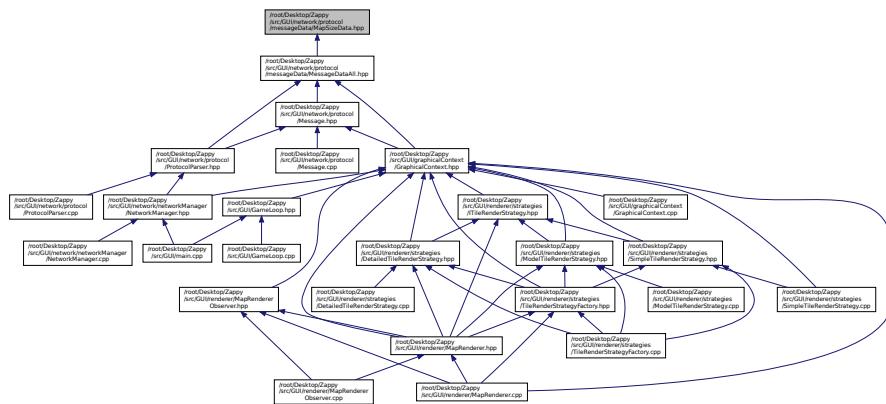
14.85 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/Map← SizeData.hpp File Reference

```
#include "IMessageData.hpp"
```

Include dependency graph for MapSizeData.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [MapSizeData](#)

14.86 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/MessageDataAll.hpp File Reference

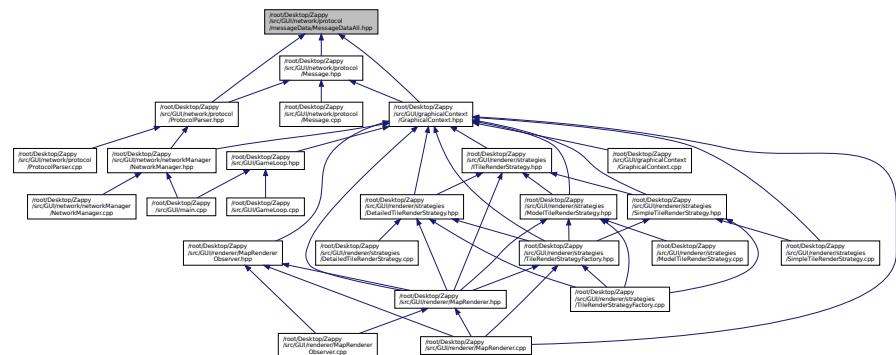
```

#include "IMessageData.hpp"
#include "MapSizeData.hpp"
#include "TileContentData.hpp"
#include "PlayerInfoData.hpp"
#include "PlayerInventoryData.hpp"
#include "BroadcastData.hpp"
#include "IncantationData.hpp"
#include "IncantationEndData.hpp"
#include "EggData.hpp"
#include "TimeUnitData.hpp"
#include "EndGameData.hpp"
#include "ServerMessageData.hpp"
#include "TeamNameData.hpp"
  
```

```
#include "ResourceData.hpp"
Include dependency graph for MessageDataAll.hpp:
```

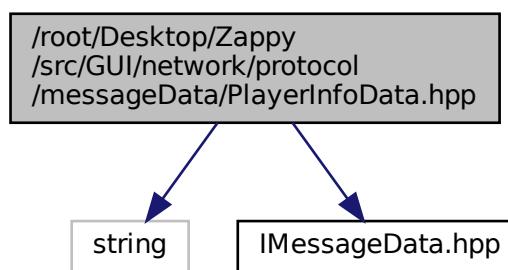


This graph shows which files directly or indirectly include this file:

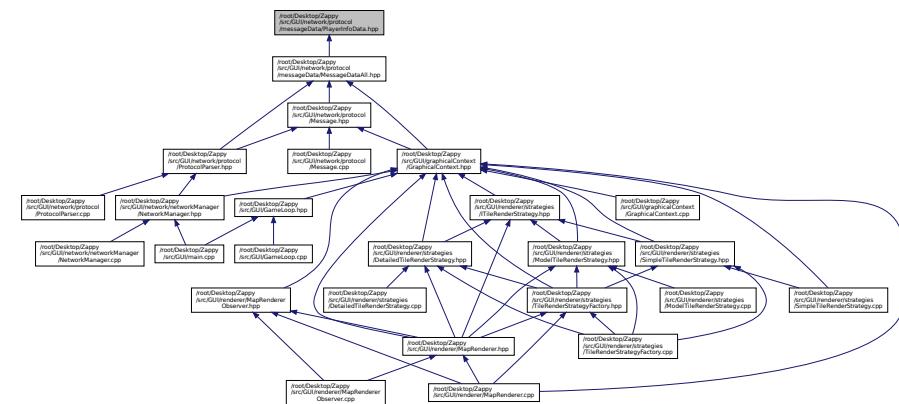


14.87 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/← PlayerInfoData.hpp File Reference

```
#include <string>
#include "IMessageData.hpp"
Include dependency graph for PlayerInfoData.hpp:
```



This graph shows which files directly or indirectly include this file:

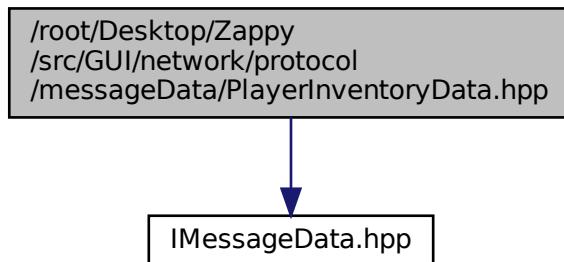


Data Structures

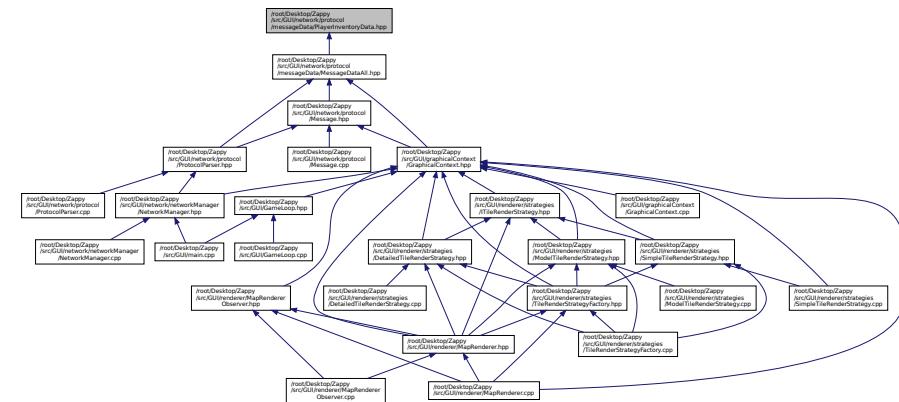
- class PlayerInfoData

14.88 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/← PlayerInventoryData.hpp File Reference

```
#include "IMessageData.hpp"
Include dependency graph for PlayerInventoryData.hpp:
```



This graph shows which files directly or indirectly include this file:

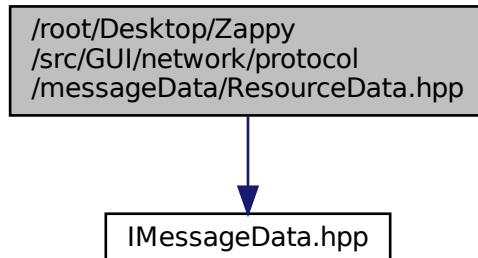


Data Structures

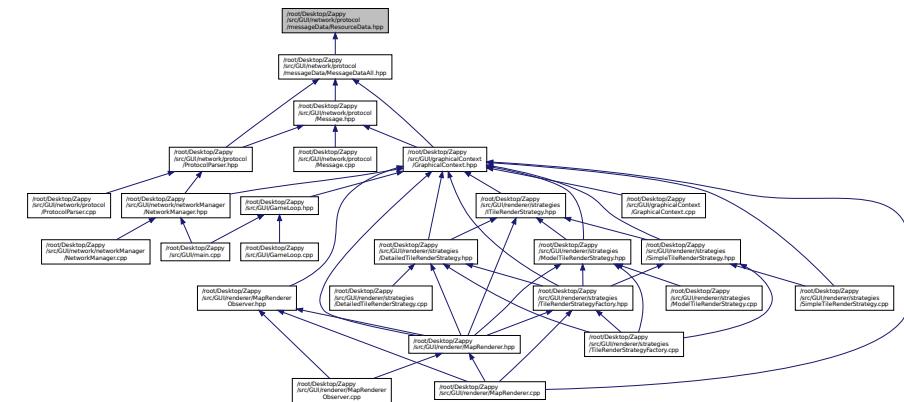
- class PlayerInventoryData

14.89 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/ResourceData.hpp File Reference

```
#include "IMessageData.hpp"  
Include dependency graph for ResourceData.hpp:
```



This graph shows which files directly or indirectly include this file:

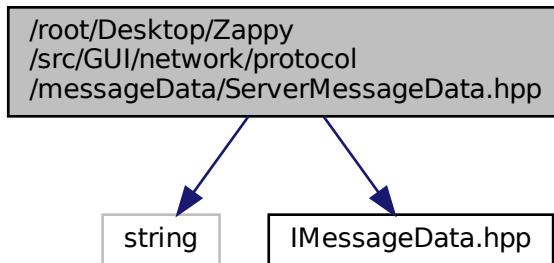


Data Structures

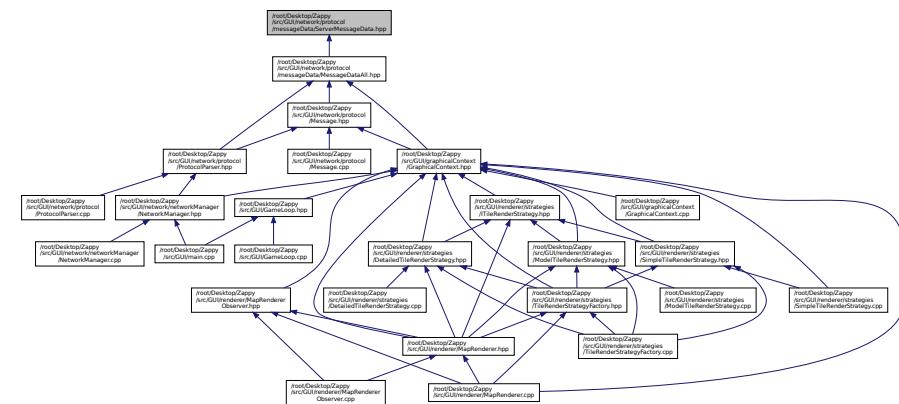
- class [ResourceData](#)

14.90 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/ServerMessageData.hpp File Reference

```
#include <string>
#include "IMessageData.hpp"
Include dependency graph for ServerMessageData.hpp:
```



This graph shows which files directly or indirectly include this file:

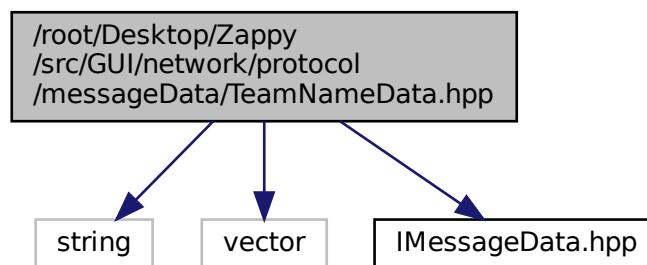


Data Structures

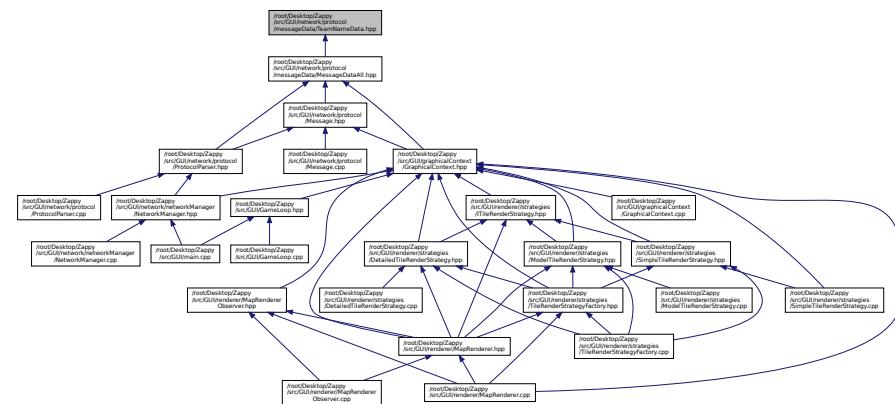
- class [ServerMessageData](#)

14.91 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/TeamNameData.hpp File Reference

```
#include <string>
#include <vector>
#include "IMessageData.hpp"
Include dependency graph for TeamNameData.hpp:
```



This graph shows which files directly or indirectly include this file:

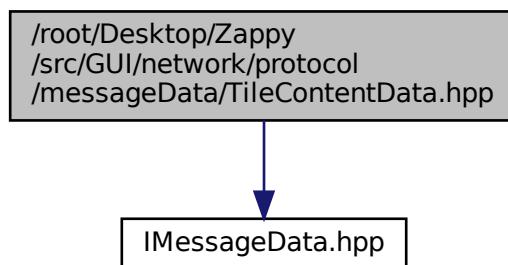


Data Structures

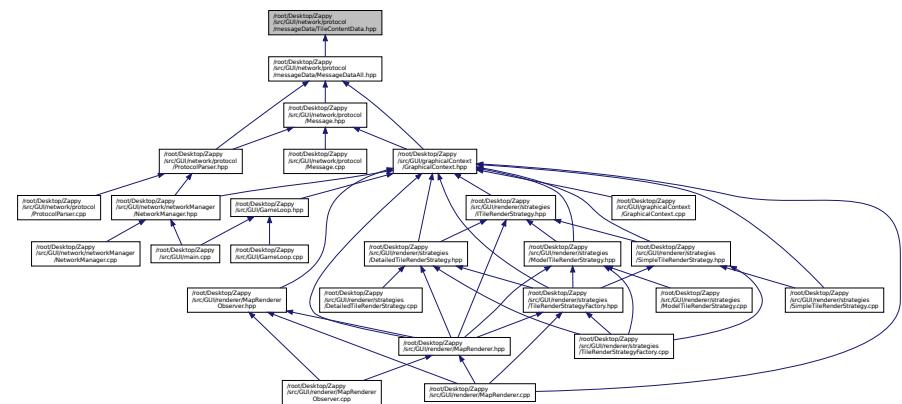
- class [TeamNameData](#)

14.92 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/TileContentData.hpp File Reference

```
#include "IMessageData.hpp"
Include dependency graph for TileContentData.hpp:
```



This graph shows which files directly or indirectly include this file:

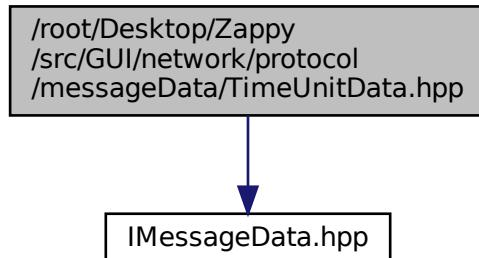


Data Structures

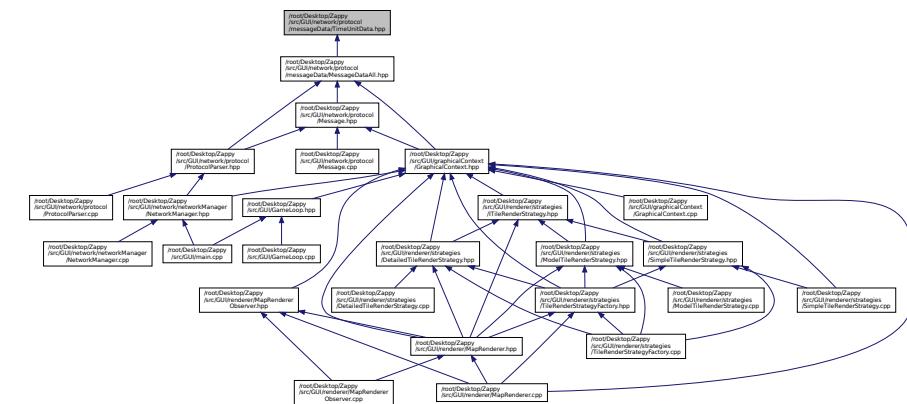
- class TileContentData

14.93 /root/Desktop/Zappy/src/GUI/network/protocol/messageData/… TimeUnitData.hpp File Reference

```
#include "IMessageData.hpp"  
Include dependency graph for TimeUnitData.hpp:
```



This graph shows which files directly or indirectly include this file:



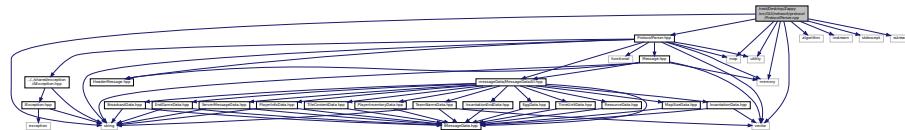
Data Structures

- class TimeUnitData

14.94 /root/Desktop/Zappy/src/GUI/network/protocol/ProtocolParser.cpp File Reference

```
#include "ProtocolParser.hpp"
#include <algorithm>
#include <iostream>
#include <map>
#include <stdexcept>
#include <vector>
#include <string>
#include <memory>
#include <utility>
#include <sstream>
```

Include dependency graph for ProtocolParser.cpp:

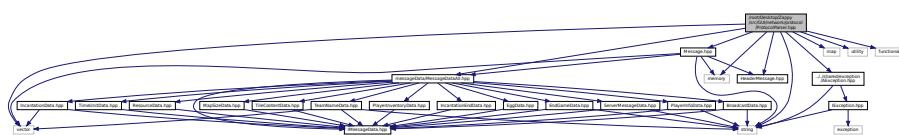


14.95 /root/Desktop/Zappy/src/GUI/network/protocol/ProtocolParser.hpp

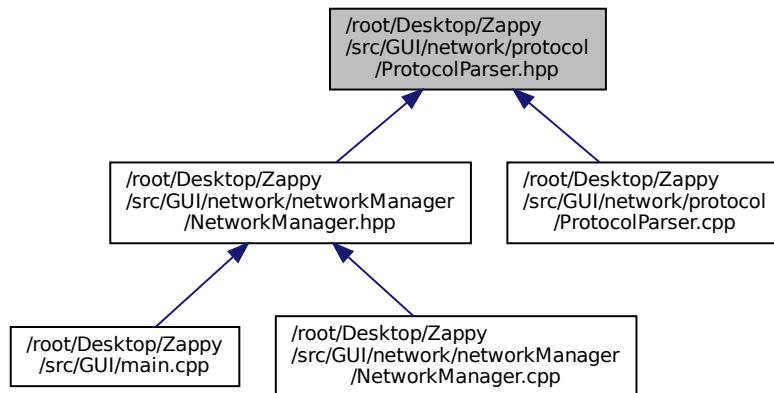
File Reference

```
#include <string>
#include <vector>
#include <map>
#include <memory>
#include <utility>
#include <functional>
#include "Message.hpp"
#include "HeaderMessage.hpp"
#include "messageData/MessageDataAll.hpp"
```

```
#include "../../shared/exception/AException.hpp"
Include dependency graph for ProtocolParser.hpp:
```



This graph shows which files directly or indirectly include this file:

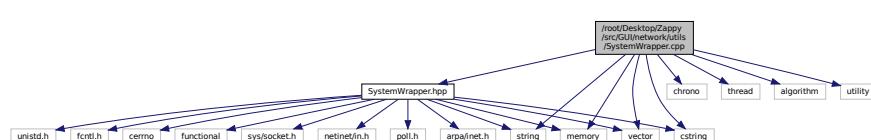


Data Structures

- class [ProtocolParser](#)

14.96 /root/Desktop/Zappy/src/GUI/network/utils/SystemWrapper.cpp File Reference

```
#include "SystemWrapper.hpp"
#include <chrono>
#include <thread>
#include <algorithm>
#include <cstring>
#include <utility>
#include <vector>
#include <memory>
#include <string>
Include dependency graph for SystemWrapper.cpp:
```



Namespaces

- `SystemWrapper`

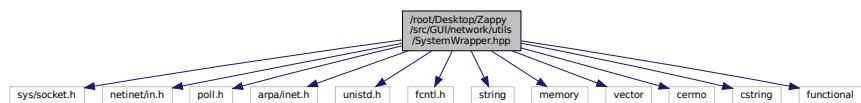
Functions

- int `SystemWrapper::createSocket` (int domain, int type, int protocol)
- int `SystemWrapper::connectSocket` (int sockfd, const `SafeSockAddr &addr`)
- int `SystemWrapper::closeSocket` (int fd)
- int `SystemWrapper::setNonBlocking` (int fd)
- `ssize_t SystemWrapper::readSocket` (int fd, std::shared_ptr< `SafeBuffer` > buffer, size_t count)
- `ssize_t SystemWrapper::writeSocket` (int fd, const `SafeBuffer &buffer`, size_t count)
- int `SystemWrapper::pollSocket` (const `SafePollFd &pfd`, int timeout)
- int `SystemWrapper::pollSocket` (const std::vector< `SafePollFd` > &pollfds, int timeout)
- bool `SystemWrapper::inetPton` (int af, const std::string &src, void *dst)
- std::string `SystemWrapper::inetNtop` (int af, const void *src)
- std::string `SystemWrapper::getErrorString` ()
- void `SystemWrapper::executeWithTimeout` (const std::function< bool()> &action, int timeoutMs)

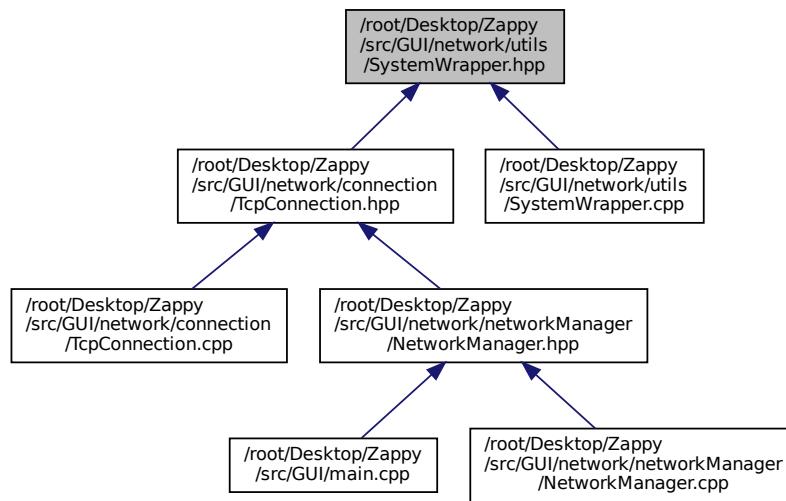
14.97 /root/Desktop/Zappy/src/GUI/network/utils/SystemWrapper.hpp File Reference

```
#include <sys/socket.h>
#include <netinet/in.h>
#include <poll.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <fcntl.h>
#include <string>
#include <memory>
#include <vector>
#include <cerrno>
#include <cstring>
#include <functional>
```

Include dependency graph for SystemWrapper.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [SystemWrapper::SafeSockAddr](#)
- class [SystemWrapper::SafePollFd](#)
- class [SystemWrapper::SafeBuffer](#)

Namespaces

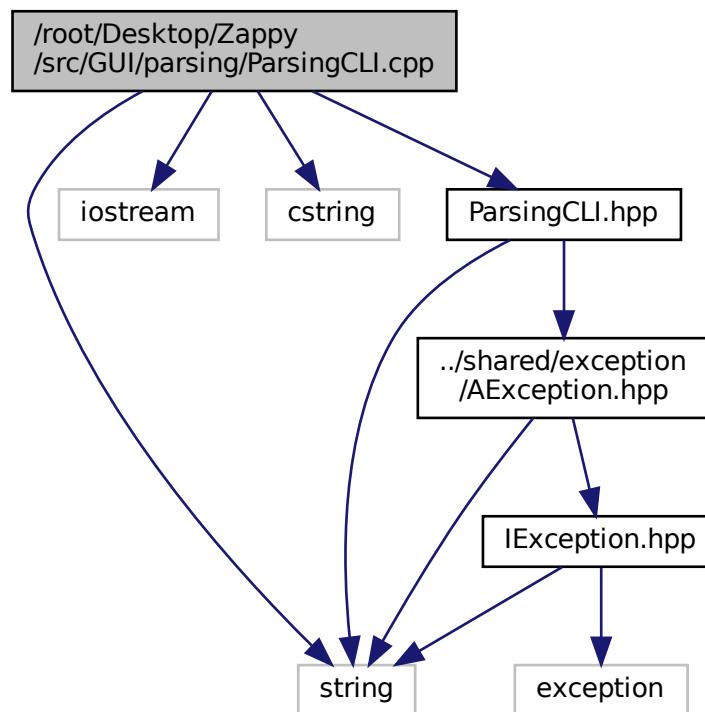
- [SystemWrapper](#)

Functions

- int [SystemWrapper::createSocket](#) (int domain, int type, int protocol)
- int [SystemWrapper::connectSocket](#) (int sockfd, const SafeSockAddr &addr)
- int [SystemWrapper::closeSocket](#) (int fd)
- int [SystemWrapper::setNonBlocking](#) (int fd)
- ssize_t [SystemWrapper::readSocket](#) (int fd, std::shared_ptr< SafeBuffer > buffer, size_t count)
- ssize_t [SystemWrapper::writeSocket](#) (int fd, const SafeBuffer &buffer, size_t count)
- int [SystemWrapper::pollSocket](#) (const SafePollFd &pfd, int timeout)
- int [SystemWrapper::pollSocket](#) (const std::vector< SafePollFd > &pollfds, int timeout)
- template<typename T>
int [SystemWrapper::getSocketOption](#) (int sockfd, int level, int optname, T *optval)
- bool [SystemWrapper::inetPton](#) (int af, const std::string &src, void *dst)
- std::string [SystemWrapper::inetNtop](#) (int af, const void *src)
- std::string [SystemWrapper::getErrorString](#) ()
- template<typename T>
std::unique_ptr< T[]> [SystemWrapper::makeUniqueArray](#) (size_t size)
- template<typename T>
std::shared_ptr< T[]> [SystemWrapper::makeSharedArray](#) (size_t size)
- void [SystemWrapper::executeWithTimeout](#) (const std::function< bool()> &action, int timeoutMs)

14.98 /root/Desktop/Zappy/src/GUI/parsing/ParsingCLI.cpp File Reference

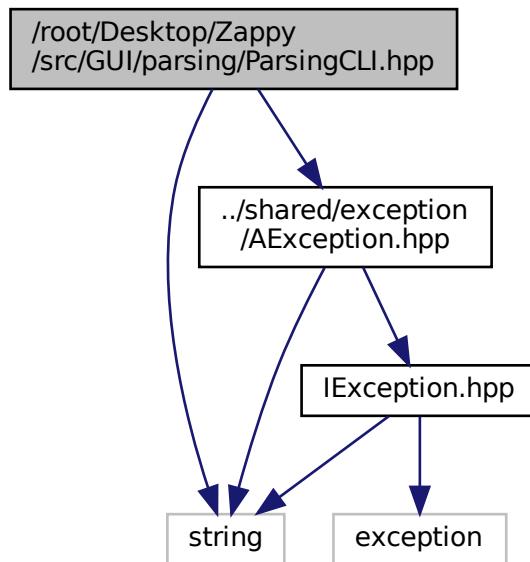
```
#include <string>
#include <iostream>
#include <cstring>
#include "ParsingCLI.hpp"
Include dependency graph for ParsingCLI.cpp:
```



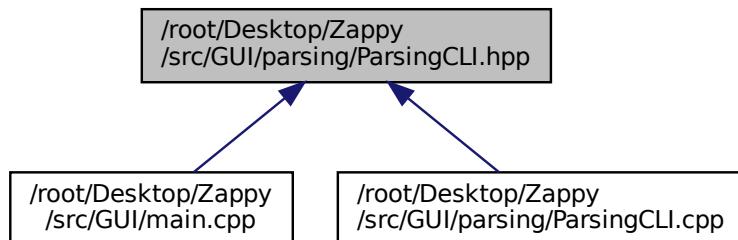
14.99 /root/Desktop/Zappy/src/GUI/parsing/ParsingCLI.hpp File Reference

```
#include <string>
#include "../shared/exception/AException.hpp"
```

Include dependency graph for ParsingCLI.hpp:



This graph shows which files directly or indirectly include this file:

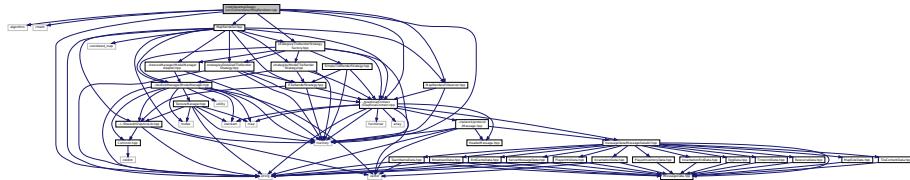


Data Structures

- class [ParsingCLI](#)

- 14.100 /root/Desktop/Zappy/src/GUI/README.md File Reference**
- 14.101 /root/Desktop/Zappy/src/Server/README.md File Reference**
- 14.102 /root/Desktop/Zappy/README.md File Reference**
- 14.103 /root/Desktop/Zappy/libs/RayGUICPP/README.md File Reference**
- 14.104 /root/Desktop/Zappy/libs/RaylibCPP/README.md File Reference**
- 14.105 /root/Desktop/Zappy/src/GUI/renderer/MapRenderer.cpp File Reference**

```
#include <algorithm>
#include <cmath>
#include <memory>
#include <string>
#include "MapRenderer.hpp"
#include "strategies/TileRenderStrategyFactory.hpp"
#include "MapRendererObserver.hpp"
#include "../graphicalContext/GraphicalContext.hpp"
Include dependency graph for MapRenderer.cpp:
```



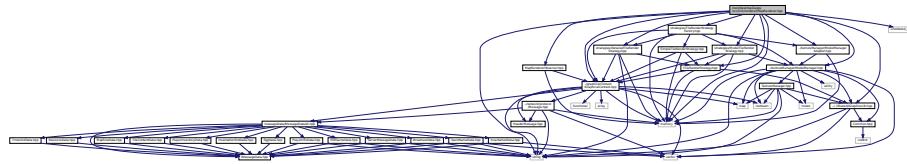
Namespaces

- [Zappy](#)

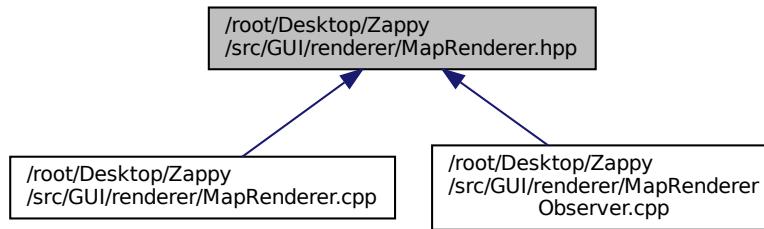
- 14.106 /root/Desktop/Zappy/src/GUI/renderer/MapRenderer.hpp File Reference**

```
#include <memory>
#include <string>
#include <unordered_map>
#include "../graphicalContext/GraphicalContext.hpp"
#include "../../Shared/IGraphicsLib.hpp"
#include "../textureManager/ModelManager.hpp"
#include "../textureManager/ModelManagerAdapter.hpp"
#include "strategies/DetailedTileRenderStrategy.hpp"
#include "strategies/ModelTileRenderStrategy.hpp"
#include "strategies/TileRenderStrategyFactory.hpp"
#include "MapRendererObserver.hpp"
#include "strategies/ITileRenderStrategy.hpp"
```

Include dependency graph for MapRenderer.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [Zappy::MapRenderer](#)

Render de carte responsable de l'affichage de la grille Implémente le pattern Bridge pour séparer l'interface du rendu.

Namespaces

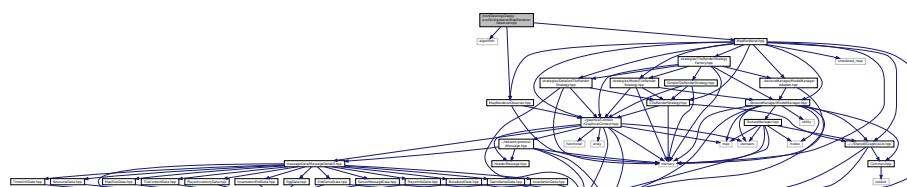
- [Zappy](#)

14.107 /root/Desktop/Zappy/src/GUI/renderer/MapRendererObserver.cpp

File Reference

```
#include <algorithm>
#include "MapRendererObserver.hpp"
#include "MapRenderer.hpp"
```

Include dependency graph for MapRendererObserver.cpp:

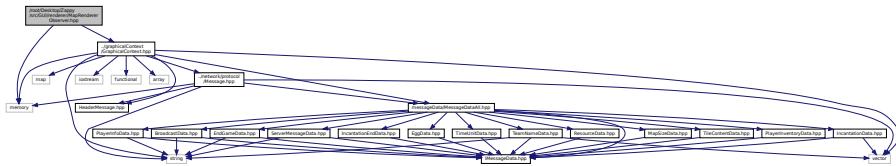


Namespaces

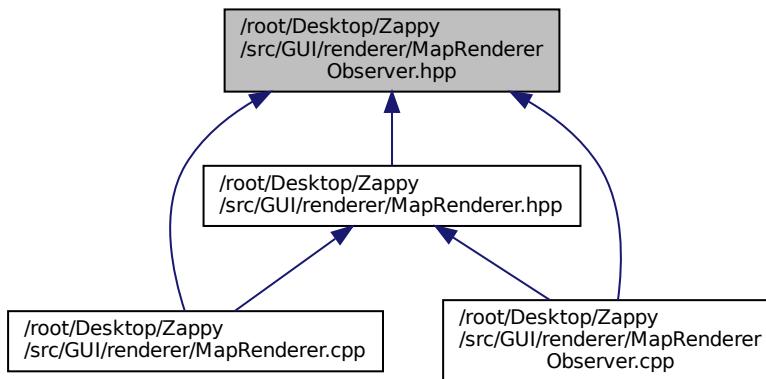
- [Zappy](#)

14.108 /root/Desktop/Zappy/src/GUI/renderer/MapRendererObserver.hpp File Reference

```
#include <memory>
#include "../graphicalContext/GraphicalContext.hpp"
Include dependency graph for MapRendererObserver.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [Zappy::MapRendererObserver](#)

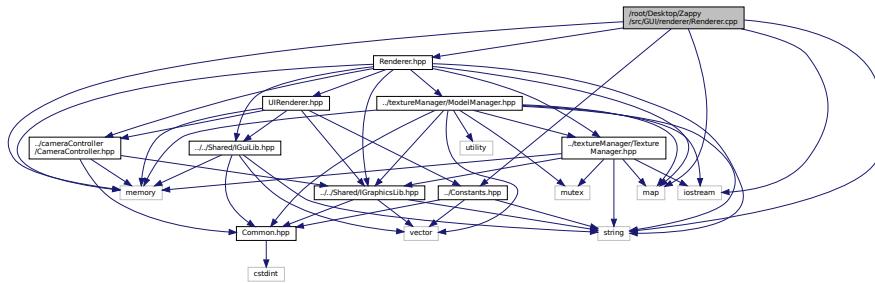
Namespaces

- [Zappy](#)

14.109 /root/Desktop/Zappy/src/GUI/renderer/Renderer.cpp File Reference

```
#include <memory>
#include <map>
#include <string>
#include <iostream>
#include "Renderer.hpp"
#include "../Constants.hpp"
```

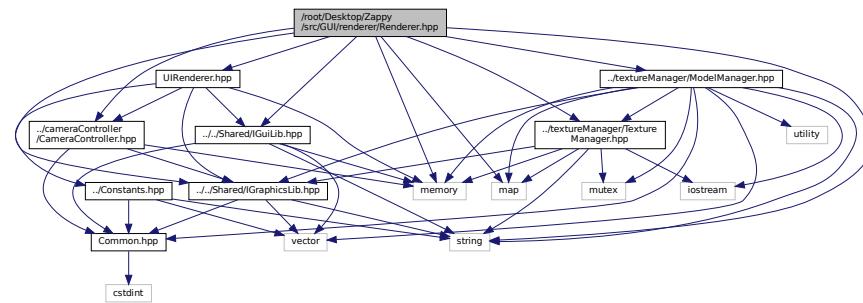
Include dependency graph for Renderer.cpp:



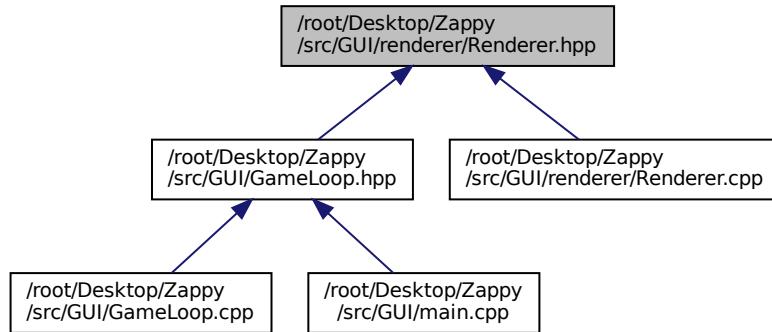
14.110 /root/Desktop/Zappy/src/GUI/renderer/Renderer.hpp File Reference

```
#include <memory>
#include <map>
#include <string>
#include "../Shared/IGraphicsLib.hpp"
#include "../Shared/IGuiLib.hpp"
#include "cameraController/Controller.hpp"
#include "UIRenderer.hpp"
#include "textureManager/TextureManager.hpp"
#include "textureManager/ModelManager.hpp"
```

Include dependency graph for Renderer.hpp:



This graph shows which files directly or indirectly include this file:



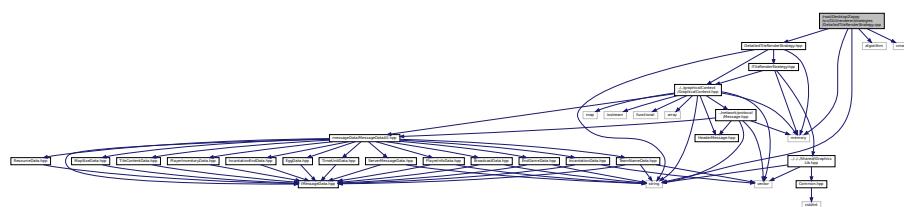
Data Structures

- class [Renderer](#)

14.111 /root/Desktop/Zappy/src/GUI/renderer/strategies/DetailedTileRenderStrategy.cpp File Reference

```
#include "DetailedTileRenderStrategy.hpp"
#include <algorithm>
#include <cmath>
#include <memory>
#include <string>
```

Include dependency graph for DetailedTileRenderStrategy.cpp:



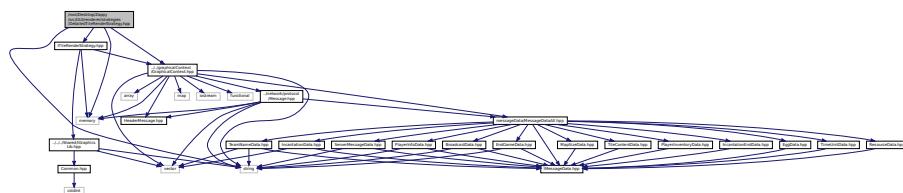
Namespaces

- [Zappy](#)

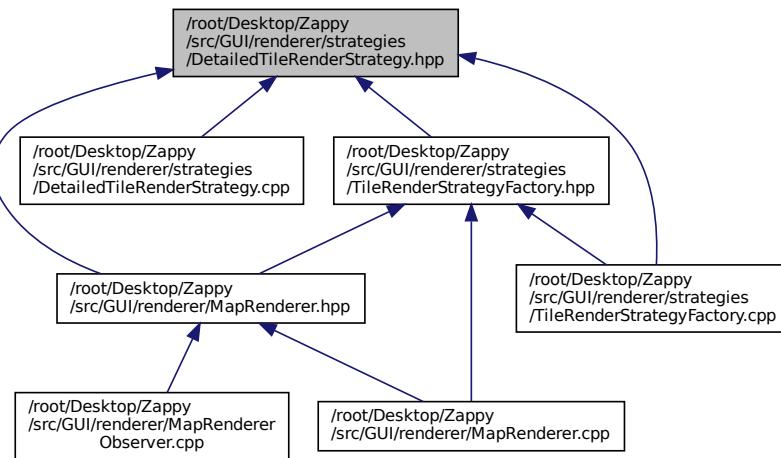
14.112 /root/Desktop/Zappy/src/GUI/renderer/strategies/DetailedTileRenderStrategy.hpp File Reference

```
#include <memory>
#include <string>
#include "ITileRenderStrategy.hpp"
#include "../../graphicalContext/GraphicalContext.hpp"
```

Include dependency graph for DetailedTileRenderStrategy.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

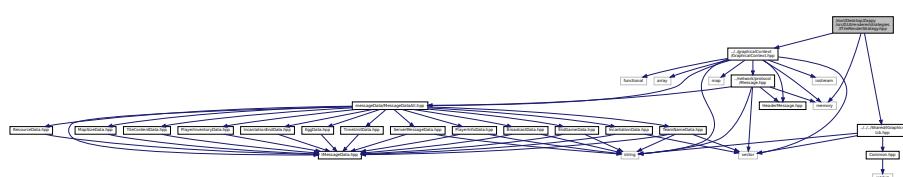
- class [Zappy::DetailedTileRenderStrategy](#)

Namespaces

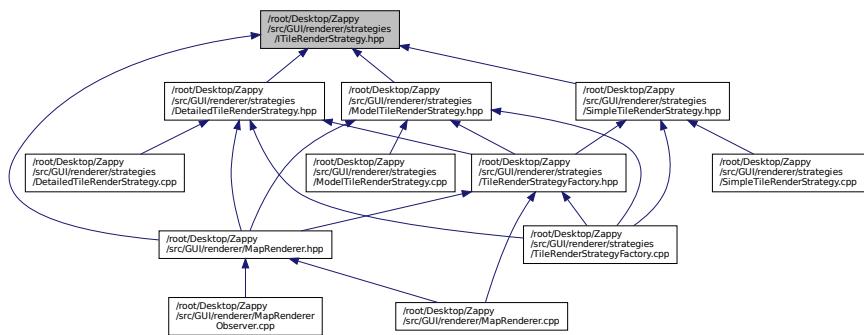
- [Zappy](#)

14.113 /root/Desktop/Zappy/src/GUI/renderer/strategies/ITileRenderStrategy.hpp File Reference

```
#include <memory>
#include "../../graphicalContext/GraphicalContext.hpp"
#include "../../../../Shared/IGraphicsLib.hpp"
Include dependency graph for ITileRenderStrategy.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

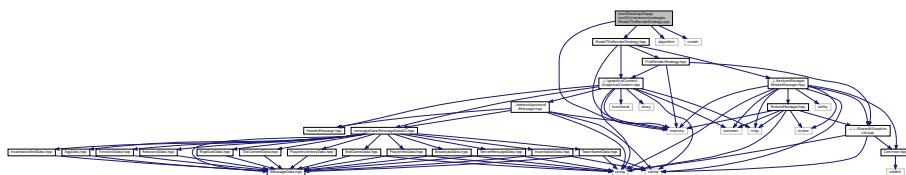
- class [Zappy::ITileRenderStrategy](#)

Namespaces

- [Zappy](#)

14.114 /root/Desktop/Zappy/src/GUI/renderer/strategies/ModelTileRenderStrategy.cpp File Reference

```
#include "ModelTileRenderStrategy.hpp"
#include <algorithm>
#include <cmath>
#include <memory>
Include dependency graph for ModelTileRenderStrategy.cpp:
```



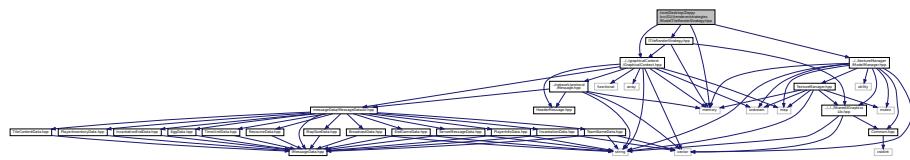
Namespaces

- [Zappy](#)

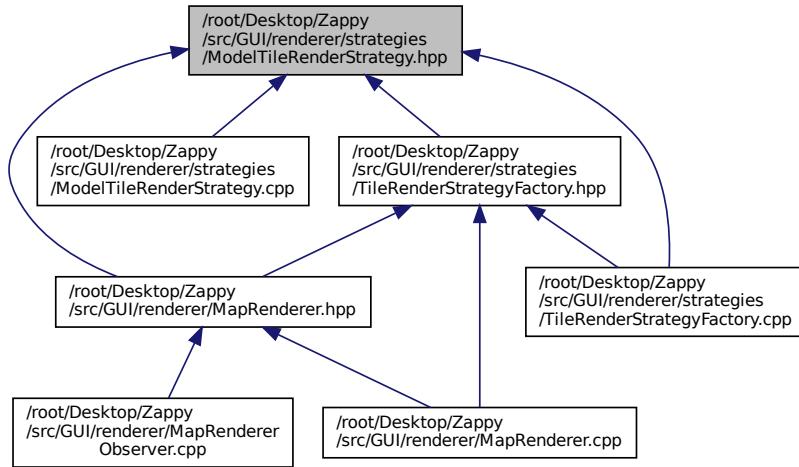
14.115 /root/Desktop/Zappy/src/GUI/renderer/strategies/ModelTileRenderStrategy.hpp File Reference

```
#include <memory>
#include "ITileRenderStrategy.hpp"
#include "../../graphicalContext/GraphicalContext.hpp"
#include "../../textureManager/ModelManager.hpp"
```

Include dependency graph for ModelTileRenderStrategy.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [Zappy::ModelTileRenderStrategy](#)

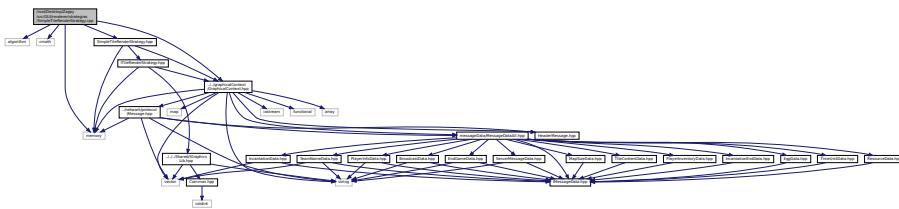
Namespaces

- [Zappy](#)

14.116 /root/Desktop/Zappy/src/GUI/renderer/strategies/SimpleTileRenderStrategy.cpp File Reference

```
#include <algorithm>
#include <cmath>
#include <memory>
#include "SimpleTileRenderStrategy.hpp"
#include "../../graphicalContext/GraphicalContext.hpp"
```

Include dependency graph for SimpleTileRenderStrategy.cpp:

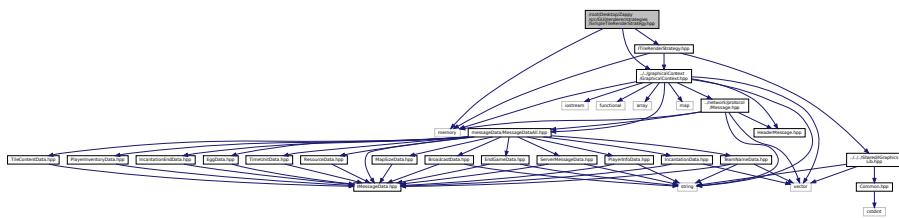


Namespaces

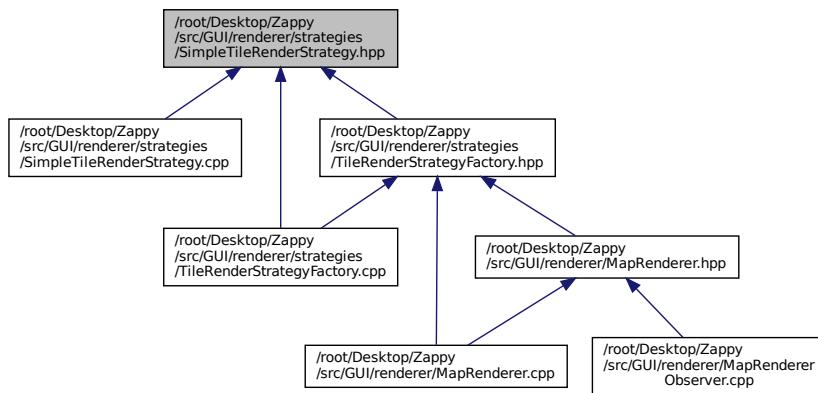
- Zappy

14.117 /root/Desktop/Zappy/src/GUI/renderer/strategies/SimpleTileRenderStrategy.hpp File Reference

```
#include <memory>
#include "ITileRenderStrategy.hpp"
#include "../../graphicalContext/GraphicalContext.hpp"
Include dependency graph for SimpleTileRenderStrategy.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [Zappy::SimpleTileRenderStrategy](#)

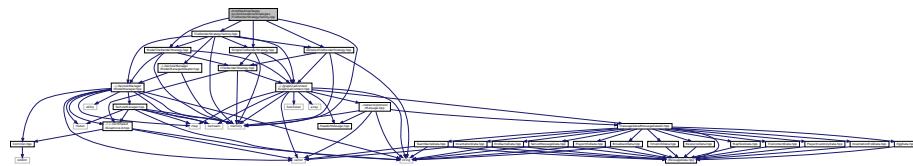
Namespaces

- [Zappy](#)

14.118 /root/Desktop/Zappy/src/GUI/renderer/strategies/TileRenderStrategyFactory.cpp File Reference

```
#include <memory>
#include "TileRenderStrategyFactory.hpp"
#include "SimpleTileRenderStrategy.hpp"
#include "ModelTileRenderStrategy.hpp"
#include "DetailedTileRenderStrategy.hpp"
```

Include dependency graph for TileRenderStrategyFactory.cpp:



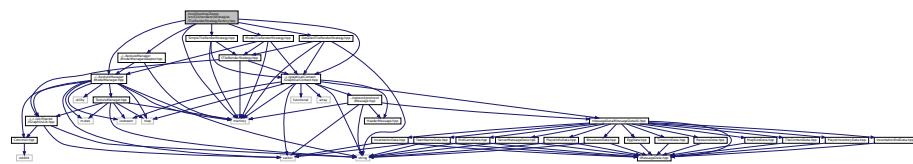
Namespaces

- [Zappy](#)

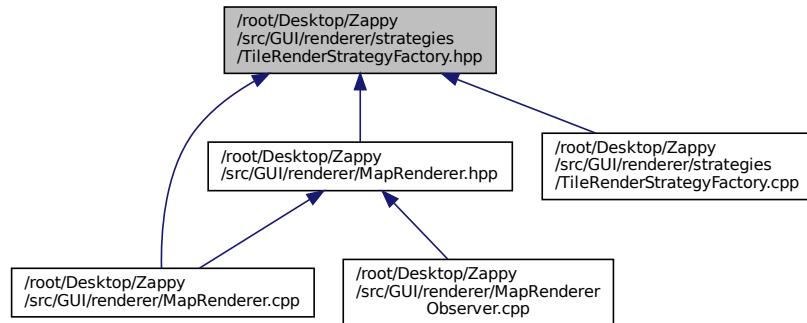
14.119 /root/Desktop/Zappy/src/GUI/renderer/strategies/TileRenderStrategyFactory.hpp File Reference

```
#include <memory>
#include "../../textureManager/ModelManager.hpp"
#include "../../graphicalContext/GraphicalContext.hpp"
#include "SimpleTileRenderStrategy.hpp"
#include "ModelTileRenderStrategy.hpp"
#include "DetailedTileRenderStrategy.hpp"
#include "../../textureManager/ModelManagerAdapter.hpp"
```

Include dependency graph for TileRenderStrategyFactory.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [Zappy::TileRenderStrategyFactory](#)

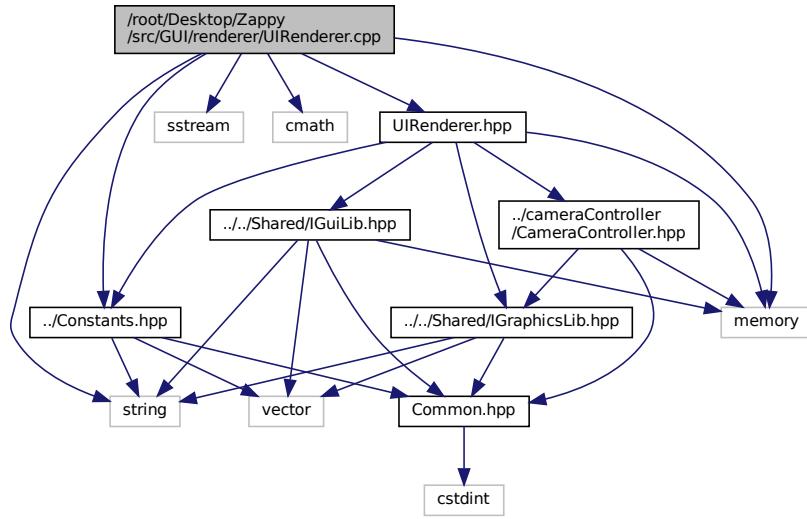
Namespaces

- [Zappy](#)

14.120 /root/Desktop/Zappy/src/GUI/renderer/UIRenderer.cpp File Reference

```
#include <string>
#include <memory>
#include <sstream>
#include <cmath>
#include "UIRenderer.hpp"
#include "../Constants.hpp"
```

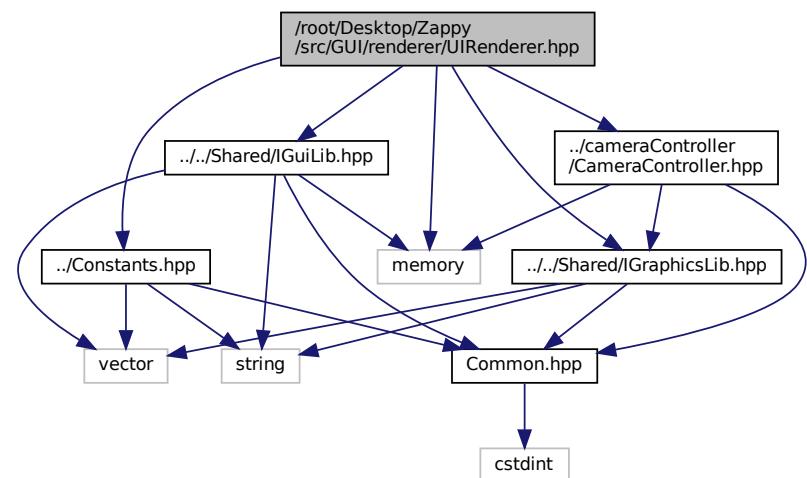
Include dependency graph for UIRenderer.cpp:



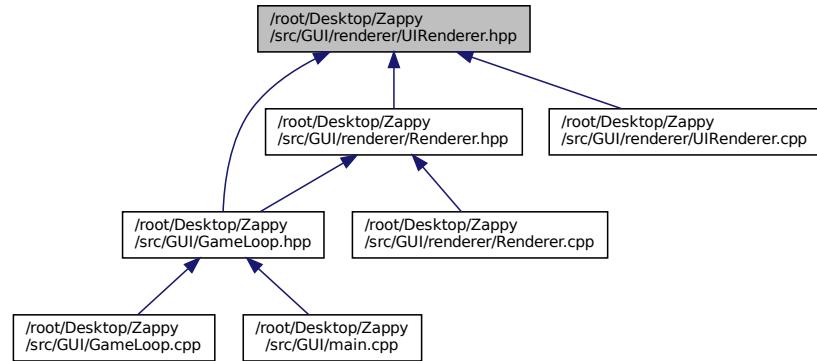
14.121 /root/Desktop/Zappy/src/GUI/renderer/UIRenderer.hpp File Reference

```
#include <memory>
#include "../Shared/IGraphicsLib.hpp"
#include "../Shared/IGuiLib.hpp"
#include "../cameraController/CameraController.hpp"
#include "../Constants.hpp"
```

Include dependency graph for UIRenderer.hpp:



This graph shows which files directly or indirectly include this file:

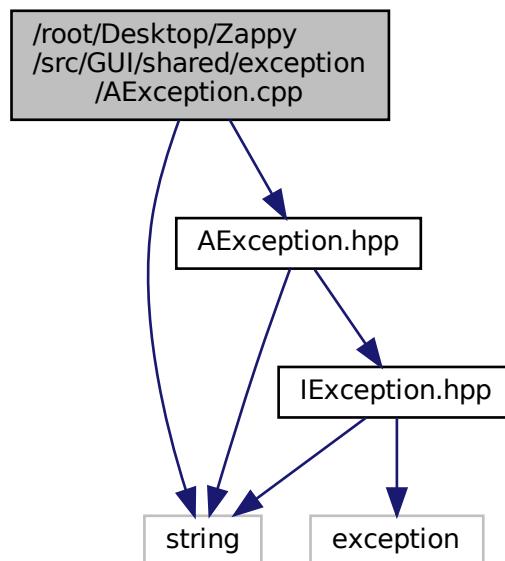


Data Structures

- class [UIRenderer](#)

14.122 /root/Desktop/Zappy/src/GUI/shared/exception/AException.cpp File Reference

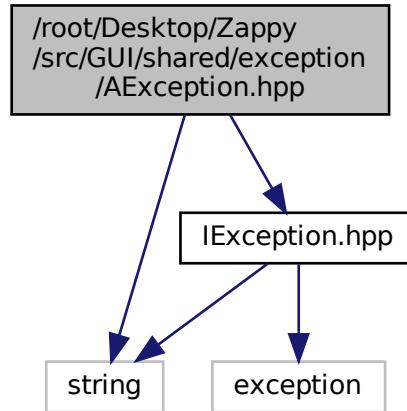
```
#include <string>
#include "AException.hpp"
Include dependency graph for AException.cpp:
```



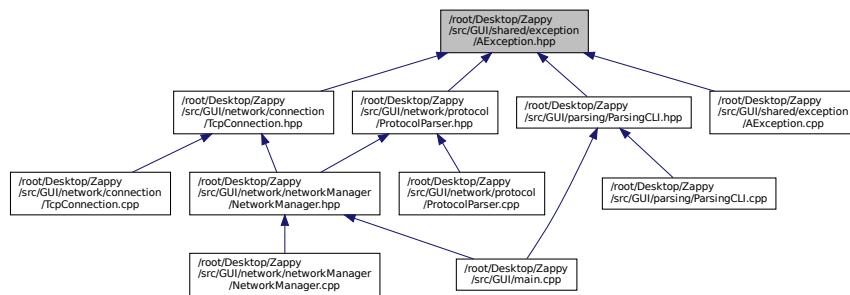
14.123 /root/Desktop/Zappy/src/GUI/shared/exception/AException.hpp

File Reference

```
#include <string>
#include "IException.hpp"
Include dependency graph for AException.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

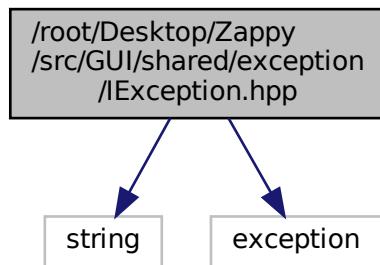
- class [AException](#)

14.124 /root/Desktop/Zappy/src/GUI/shared/exception/IException.hpp

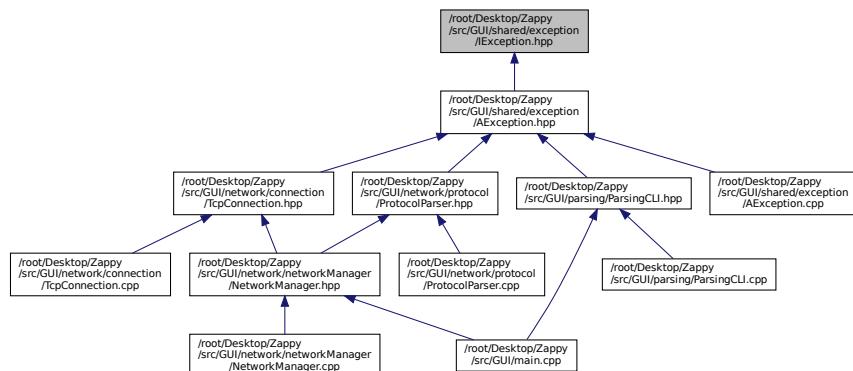
File Reference

```
#include <string>
#include <exception>
```

Include dependency graph for `IException.hpp`:



This graph shows which files directly or indirectly include this file:



Data Structures

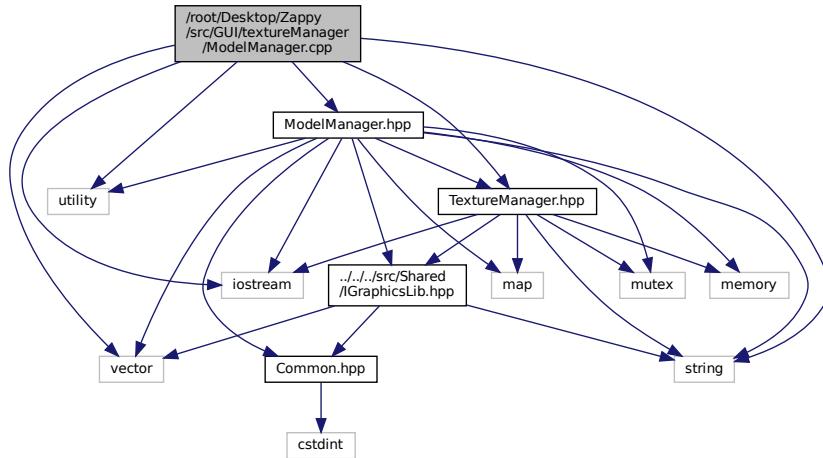
- class `IException`

14.125 /root/Desktop/Zappy/src/GUI/textureManager/ModelManager.cpp File Reference

```

#include <iostream>
#include <vector>
#include <string>
#include <utility>
#include "ModelManager.hpp"
#include "TextureManager.hpp"
  
```

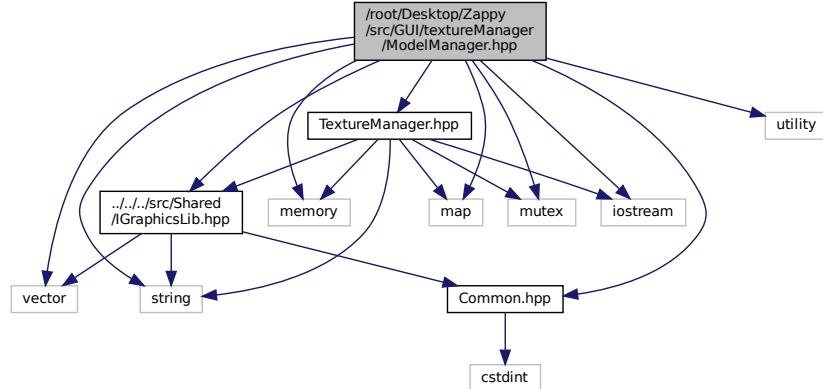
Include dependency graph for ModelManager.cpp:



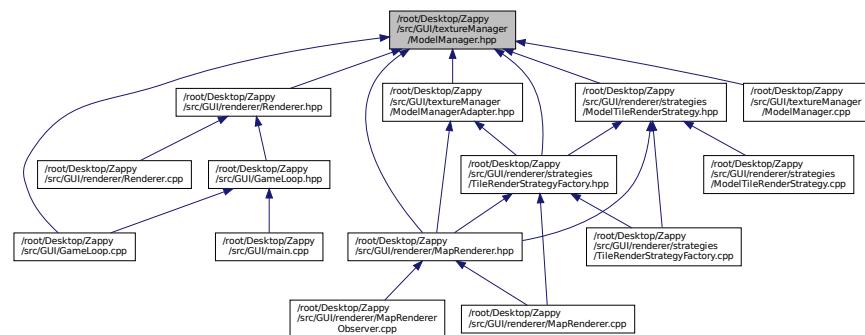
14.126 /root/Desktop/Zappy/src/GUI/textureManager/ModelManager.hpp File Reference

```
#include <string>
#include <memory>
#include <map>
#include <mutex>
#include <iostream>
#include <utility>
#include <vector>
#include "../..../src/Shared/IGraphicsLib.hpp"
#include "../..../src/Shared/Common.hpp"
#include "TextureManager.hpp"
```

Include dependency graph for ModelManager.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Model3D**

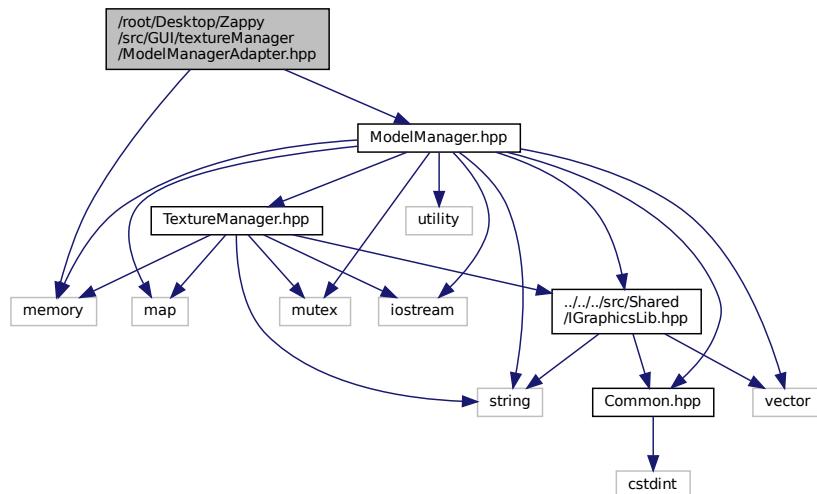
Structure représentant un modèle 3D avec ses textures associées.

- class **ModelManager**

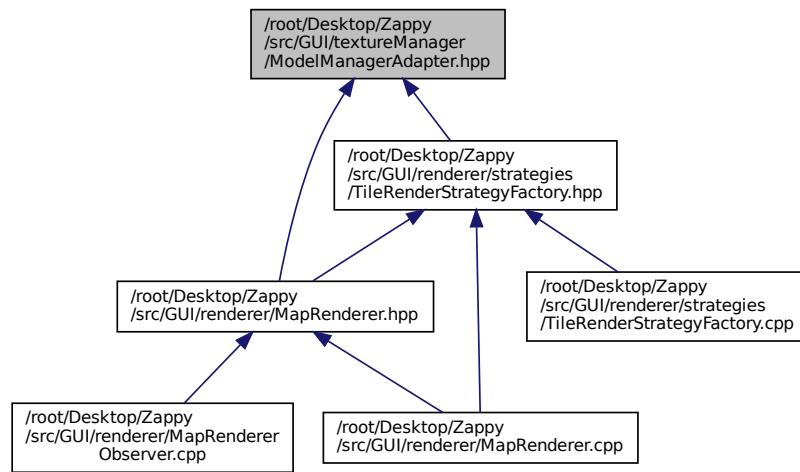
Gestionnaire de modèles 3D singleton qui implémente un pattern de pool de ressources pour éviter de charger plusieurs fois les mêmes modèles.

14.127 /root/Desktop/Zappy/src/GUI/textureManager/ModelManagerAdapter.hpp File Reference

```
#include <memory>
#include "ModelManager.hpp"
Include dependency graph for ModelManagerAdapter.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [Zappy::ModelManagerAdapter](#)

Adaptateur qui permet de gérer [ModelManager](#) comme un std::shared_ptr. Cette classe encapsule le singleton [ModelManager](#) pour l'utiliser avec des shared_ptr.

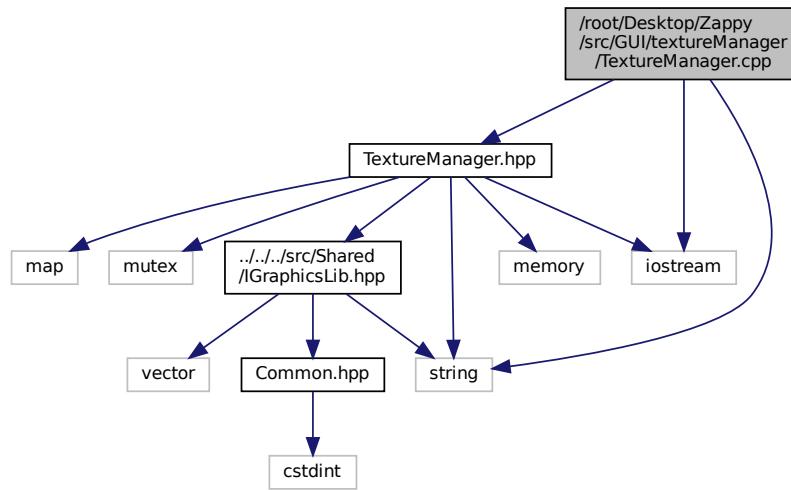
Namespaces

- [Zappy](#)

14.128 /root/Desktop/Zappy/src/GUI/textureManager/TextureManager.cpp File Reference

```
#include "TextureManager.hpp"
#include <iostream>
#include <string>
```

Include dependency graph for TextureManager.cpp:

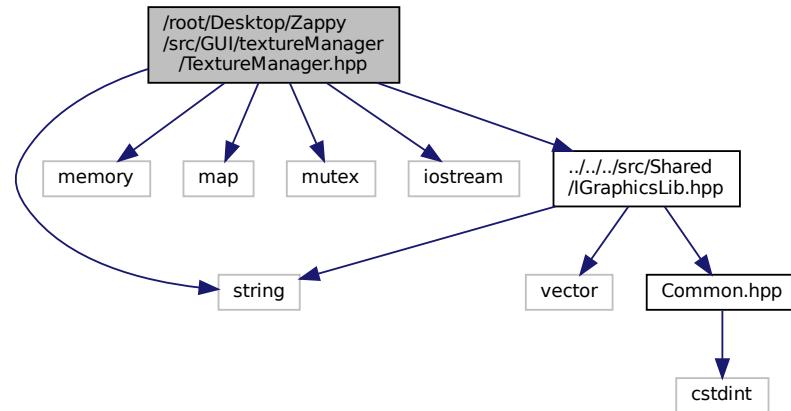


14.129 /root/Desktop/Zappy/src/GUI/textureManager/TextureManager.hpp File Reference

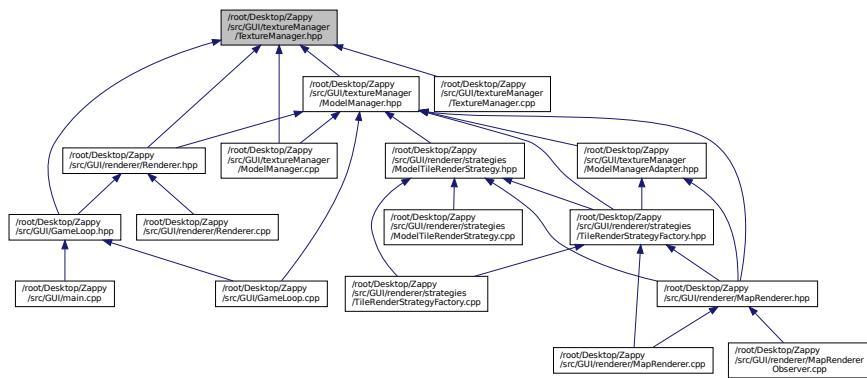
```

#include <string>
#include <memory>
#include <map>
#include <mutex>
#include <iostream>
#include "../../../../src/Shared/IGraphicsLib.hpp"
  
```

Include dependency graph for TextureManager.hpp:



This graph shows which files directly or indirectly include this file:



Data Structures

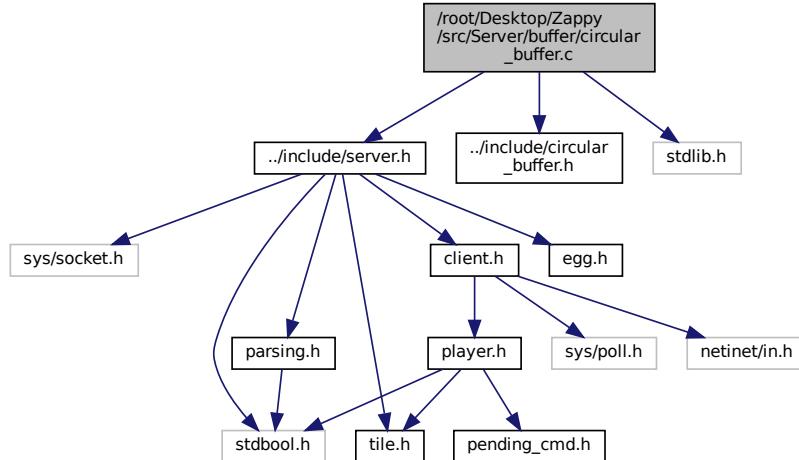
- class [TextureManager](#)

Gestionnaire de textures singleton qui implémente un pattern de pool de ressources pour éviter de charger plusieurs fois les mêmes textures.

14.130 /root/Desktop/Zappy/src/Server/buffer/circular_buffer.c File Reference

```
#include "../include/server.h"
#include "../include/circular_buffer.h"
#include <stdlib.h>
```

Include dependency graph for circular_buffer.c:



Functions

- void [init_circular_buffer](#) ([circular_buffer_t](#) *cb)
- int [add_to_circular_buffer](#) ([circular_buffer_t](#) *cb, char byte)

- int `find_command_end (circular_buffer_t *cb)`
- char * `extract_command (circular_buffer_t *cb, int cmd_length)`

14.130.1 Function Documentation

14.130.1.1 add_to_circular_buffer()

```
int add_to_circular_buffer (
    circular_buffer_t * cb,
    char byte )
```

Definition at line 18 of file circular_buffer.c.

```
19 {
20     if (cb->count >= BUFFER_SIZE - 1)
21         return -1;
22     cb->buffer[cb->end] = byte;
23     cb->end = (cb->end + 1) % BUFFER_SIZE;
24     cb->count++;
25     return 0;
26 }
```

14.130.1.2 extract_command()

```
char* extract_command (
    circular_buffer_t * cb,
    int cmd_length )
```

Definition at line 45 of file circular_buffer.c.

```
46 {
47     char *command = malloc(cmd_length + 1);
48
49     if (!command)
50         return NULL;
51     for (int i = 0; i < cmd_length; i++) {
52         command[i] = cb->buffer[cb->start];
53         cb->start = (cb->start + 1) % BUFFER_SIZE;
54         cb->count--;
55     }
56     command[cmd_length] = '\0';
57     return command;
58 }
```

14.130.1.3 find_command_end()

```
int find_command_end (
    circular_buffer_t * cb )
```

Definition at line 28 of file circular_buffer.c.

```
29 {
30     int pos = cb->start;
31
32     if (cb->count < 1)
33         return -1;
34     for (int i = 0; i < cb->count; i++) {
35         if ((cb->buffer[pos] == '\r' && cb->buffer[pos + 1] == '\n')
36             || (cb->buffer[pos] == '\n' && cb->buffer[pos + 1] == '\0'))
37             return -1;
38         if (cb->buffer[pos] == '\n')
39             return i + 1;
40         pos = (pos + 1) % BUFFER_SIZE;
41     }
42     return -1;
43 }
```

14.130.1.4 init_circular_buffer()

```
void init_circular_buffer (
    circular_buffer_t * cb )
```

Definition at line 11 of file circular_buffer.c.

```

12 {
13     cb->start = 0;
14     cb->end = 0;
15     cb->count = 0;
16 }

```

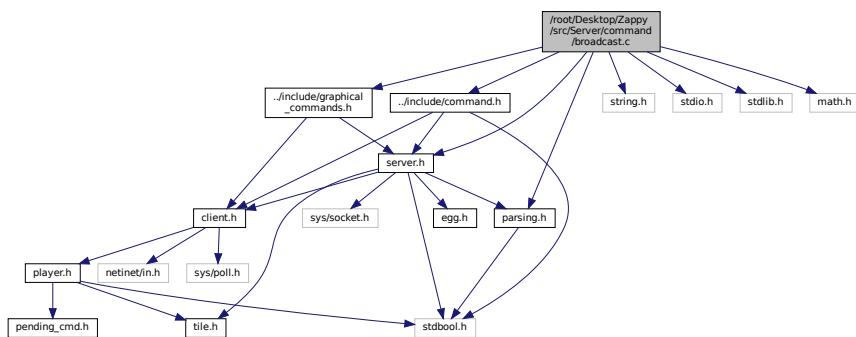
14.131 /root/Desktop/Zappy/src/Server/command/broadcast.c File Reference

```

#include "../include/command.h"
#include "../include/graphical_commands.h"
#include "../include/server.h"
#include "../include/parsing.h"
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

```

Include dependency graph for broadcast.c:



Functions

- static int `calculate_shortest_distance_x` (int sender_x, int receiver_x, int width)
- static int `calculate_shortest_distance_y` (int sender_y, int receiver_y, int height)
- static int `get_base_direction` (int dx, int dy)
- static int `calculate_direction` (client_t *receiver, int dx, int dy)
- static void `send_broadcast_to_client` (server_t *server, client_t *sender, client_t *receiver, char *message)
- static int `calculate_message_length` (char **buffer, int arr_length)
- static void `concatenate_message_parts` (char **buffer, int arr_length, char *message)
- static char * `build_broadcast_message` (char **buffer)
- void `broadcast` (server_t *server, client_t *user, char **buffer)

14.131.1 Function Documentation

14.131.1.1 broadcast()

```

void broadcast (
    server_t * server,
    client_t * user,
    char ** buffer )

```

Definition at line 149 of file broadcast.c.

```

150 {
151     char *message;

```

```

152     client_t *current;
153
154     if (!user || !user->player || !buffer || arr_len(buffer) < 2)
155         return write_command_output(user->client_fd, "ko\n");
156     message = build_broadcast_message(buffer);
157     if (!message)
158         return write_command_output(user->client_fd, "ko\n");
159     current = server->client;
160     if (current)
161         current = current->next;
162     command_pbc(server, user, message);
163     while (current) {
164         if (current->player && current != user && current->type != GRAPHICAL)
165             send_broadcast_to_client(server, user, current, message);
166         current = current->next;
167     }
168     free(message);
169     write_command_output(user->client_fd, "ok\n");
170 }
```

14.131.1.2 build_broadcast_message()

```
static char* build_broadcast_message (
    char ** buffer ) [static]
```

Definition at line 133 of file broadcast.c.

```

134 {
135     int arr_length = arr_len(buffer);
136     int total_len;
137     char *message;
138
139     if (!buffer || arr_length < 2)
140         return NULL;
141     total_len = calculate_message_length(buffer, arr_length);
142     message = malloc(total_len);
143     if (!message)
144         return NULL;
145     concatenate_message_parts(buffer, arr_length, message);
146     return message;
147 }
```

14.131.1.3 calculate_direction()

```
static int calculate_direction (
    client_t * receiver,
    int dx,
    int dy ) [static]
```

Definition at line 69 of file broadcast.c.

```

70 {
71     int base_direction;
72
73     if (dx == 0 && dy == 0)
74         return 0;
75     base_direction = get_base_direction(dx, dy);
76     if (base_direction == 0)
77         return 0;
78     return ((base_direction - 1 + 2 * receiver->player->rotation) % 8) + 1;
79 }
```

14.131.1.4 calculate_message_length()

```
static int calculate_message_length (
    char ** buffer,
    int arr_length ) [static]
```

Definition at line 104 of file broadcast.c.

```

105 {
106     int total_len = 0;
107
108     for (int i = 1; i < arr_length; i++) {
109         total_len += strlen(buffer[i]);
110         if (i > 1)
111             total_len += 1;
112     }
113     total_len += 1;
```

```
114     return total_len;
115 }
```

14.131.1.5 calculate_shortest_distance_x()

```
static int calculate_shortest_distance_x (
    int sender_x,
    int receiver_x,
    int width ) [static]
```

Definition at line 16 of file broadcast.c.

```
18 {
19     int direct_distance = receiver_x - sender_x;
20     int wrap_distance_right = (receiver_x + width) - sender_x;
21     int wrap_distance_left = receiver_x - (sender_x + width);
22
23     if (abs(direct_distance) <= abs(wrap_distance_right) &&
24         abs(direct_distance) <= abs(wrap_distance_left))
25         return direct_distance;
26     if (abs(wrap_distance_right) <= abs(wrap_distance_left))
27         return wrap_distance_right;
28     return wrap_distance_left;
29 }
```

14.131.1.6 calculate_shortest_distance_y()

```
static int calculate_shortest_distance_y (
    int sender_y,
    int receiver_y,
    int height ) [static]
```

Definition at line 31 of file broadcast.c.

```
33 {
34     int direct_distance = receiver_y - sender_y;
35     int wrap_distance_down = (receiver_y + height) - sender_y;
36     int wrap_distance_up = receiver_y - (sender_y + height);
37
38     if (abs(direct_distance) <= abs(wrap_distance_down) &&
39         abs(direct_distance) <= abs(wrap_distance_up))
40         return direct_distance;
41     if (abs(wrap_distance_down) <= abs(wrap_distance_up))
42         return wrap_distance_down;
43     return wrap_distance_up;
44 }
```

14.131.1.7 concatenate_message_parts()

```
static void concatenate_message_parts (
    char ** buffer,
    int arr_length,
    char * message ) [static]
```

Definition at line 117 of file broadcast.c.

```
119 {
120     int current_pos = 0;
121
122     message[0] = '\0';
123     for (int i = 1; i < arr_length; i++) {
124         if (i > 1) {
125             strcpy(message + current_pos, " ");
126             current_pos += 1;
127         }
128         strcpy(message + current_pos, buffer[i]);
129         current_pos += strlen(buffer[i]);
130     }
131 }
```

14.131.1.8 get_base_direction()

```
static int get_base_direction (
    int dx,
```

```
        int dy ) [static]
Definition at line 46 of file broadcast.c.
```

```
47 {
48     if (dx == 0 && dy == 0)
49         return 0;
50     if (dx > 0 && dy == 0)
51         return 1;
52     if (dx > 0 && dy < 0)
53         return 2;
54     if (dx == 0 && dy < 0)
55         return 3;
56     if (dx < 0 && dy < 0)
57         return 4;
58     if (dx < 0 && dy == 0)
59         return 5;
60     if (dx < 0 && dy > 0)
61         return 6;
62     if (dx == 0 && dy > 0)
63         return 7;
64     if (dx > 0 && dy > 0)
65         return 8;
66     return 8;
67 }
```

14.131.1.9 send_broadcast_to_client()

```
static void send_broadcast_to_client (
    server_t * server,
    client_t * sender,
    client_t * receiver,
    char * message ) [static]
```

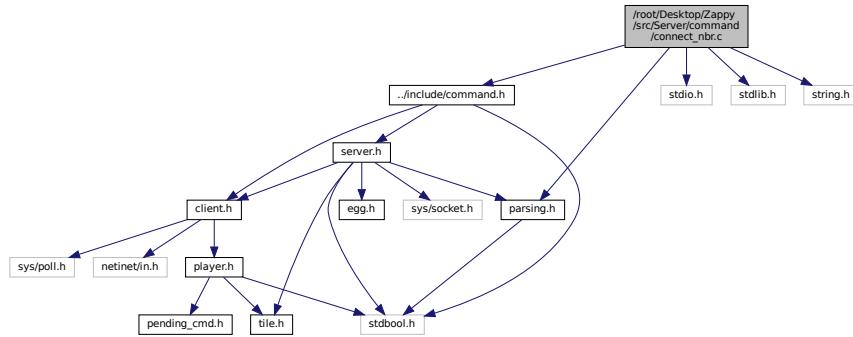
Definition at line 81 of file broadcast.c.

```
83 {
84     int dx = 0;
85     int dy = 0;
86     int direction = 0;
87     size_t res_size = snprintf(NULL, 0, "message %d, %s\n", 0, message) + 1;
88     char *response = calloc(res_size, sizeof(char));
89
90     if (!receiver || !receiver->player || receiver == sender)
91         return;
92     if (response == NULL)
93         server_err("Malloc failed for allocating response for broadcast");
94     dx = calculate_shortest_distance_x(sender->player->pos_x,
95                                         receiver->player->pos_x, server->parsed_info->width);
96     dy = calculate_shortest_distance_y(sender->player->pos_y,
97                                         receiver->player->pos_y, server->parsed_info->height);
98     direction = calculate_direction(receiver, dx, dy);
99     snprintf(response, res_size, "message %d, %s\n", direction, message);
100    write_command_output(receiver->client_fd, response);
101    free(response);
102 }
```

14.132 /root/Desktop/Zappy/src/Server/command/connect_nbr.c File Reference

```
#include "../include/command.h"
#include "../include/parsing.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Include dependency graph for connect_nbr.c:



Functions

- static int `count_team_eggs (server_t *server, char *team_name)`
- static int `count_team_players (server_t *server, char *team_name)`
- static void `format_response (int available_slots, client_t *client)`
- int `connect_nbr_srv (server_t *server, char *team)`
- void `connect_nbr (server_t *server, client_t *client, char **buffer)`

14.132.1 Function Documentation

14.132.1.1 connect_nbr()

```
void connect_nbr (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 64 of file connect_nbr.c.

```
65 {
66     int available_slots;
67
68     if (!client || !client->player || !client->player->team_name || 
69         arr_len(buffer) != 1) {
70         write_command_output(client->client_fd, "ko\n");
71         return;
72     }
73     available_slots = connect_nbr_srv(server, client->player->team_name);
74     format_response(available_slots, client);
75 }
```

14.132.1.2 connect_nbr_srv()

```
int connect_nbr_srv (
    server_t * server,
    char * team )
```

Definition at line 52 of file connect_nbr.c.

```
53 {
54     int team_eggs = count_team_eggs(server, team);
55     int team_players = count_team_players(server, team);
56     int max_clients = server->parsed_info->client_nb;
57     int available_slots = max_clients - (team_players + team_eggs);
58
59     if (available_slots < 0)
60         available_slots = 0;
61     return available_slots;
62 }
```

14.132.1.3 count_team_eggs()

```
static int count_team_eggs (
    server_t * server,
    char * team_name ) [static]
```

Definition at line 13 of file connect_nbr.c.

```
14 {
15     egg_t *current = server->eggs;
16     egg_t *next;
17     int count = 0;
18
19     while (current) {
20         next = current->next;
21         if (current->team_name && strcmp(current->team_name, team_name) == 0)
22             count++;
23         current = next;
24     }
25     return count;
26 }
```

14.132.1.4 count_team_players()

```
static int count_team_players (
    server_t * server,
    char * team_name ) [static]
```

Definition at line 28 of file connect_nbr.c.

```
29 {
30     client_t *current = server->client;
31     int count = 0;
32
33     while (current) {
34         if (current->player && current->player->team_name &&
35             strcmp(current->player->team_name, team_name) == 0)
36             count++;
37         current = current->next;
38     }
39     return count;
40 }
```

14.132.1.5 format_response()

```
static void format_response (
    int available_slots,
    client_t * client ) [static]
```

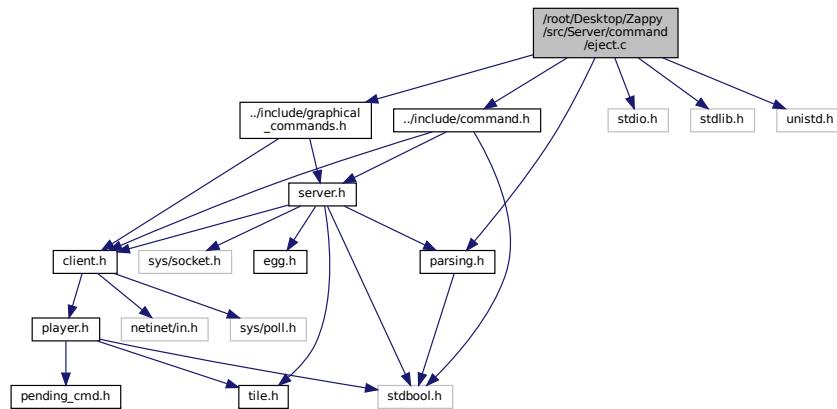
Definition at line 42 of file connect_nbr.c.

```
43 {
44     size_t res_size = snprintf(NULL, 0, "%d\n", available_slots) + 1;
45     char *response = malloc(sizeof(char) * res_size);
46
47     snprintf(response, res_size, "%d\n", available_slots);
48     write_command_output(client->client_fd, response);
49     free(response);
50 }
```

14.133 /root/Desktop/Zappy/src/Server/command/eject.c File Reference

```
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include "../include/parsing.h"
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
```

Include dependency graph for eject.c:



Functions

- static void `convert_rotation_to_vector` (`client_t` *client, float *x, float *y)
- static void `wrap_position` (`server_t` *server, int *x, int *y)
- static int `direction_push` (float x, float y)
- static char * `get_string_to_send` (float x, float y)
- static void `push_single_client` (`server_t` *server, `client_t` *client, `client_t` *tmp, char *msg)
- void `push_client` (`server_t` *server, `client_t` *client, float x, float y)
- static void `push_eggs` (`server_t` *server, int old_x, int old_y)
- void `eject` (`server_t` *server, `client_t` *client, char **buffer)

14.133.1 Function Documentation

14.133.1.1 `convert_rotation_to_vector()`

```
static void convert_rotation_to_vector (
    client_t * client,
    float * x,
    float * y ) [static]
```

Definition at line 15 of file eject.c.

```

16 {
17     switch (client->player->rotation) {
18         case RIGHT:
19             *x = 1.0;
20             *y = 0.0;
21             break;
22         case DOWN:
23             *x = 0.0;
24             *y = 1.0;
25             break;
26         case LEFT:
27             *x = -1.0;
28             *y = 0.0;
29             break;
30         case UP:
31         default:
32             *x = 0.0;
33             *y = -1.0;
34             break;
35     }
36 }
```

14.133.1.2 direction_push()

```
static int direction_push (
    float x,
    float y ) [static]
```

Definition at line 50 of file eject.c.

```
51 {
52     if (x == 1 && y == 0)
53         return 1;
54     if (x == 0 && y == 1)
55         return 7;
56     if (x == 0 && y == -1)
57         return 3;
58     if (x == -1 && y == 0)
59         return 5;
60     return -1;
61 }
```

14.133.1.3 eject()

```
void eject (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 124 of file eject.c.

```
125 {
126     float x = 0;
127     float y = 0;
128
129     if (!client || !client->player || arr_len(buffer) != 1)
130         return write_command_output(client->client_fd, "ko\n");
131     convert_rotation_to_vector(client, &x, &y);
132     push_client(server, client, x, y);
133     push_eggs(server, client->player->pos_x, client->player->pos_y);
134     command_pex(server, client);
135     write_command_output(client->client_fd, "ok\n");
136 }
```

14.133.1.4 get_string_to_send()

```
static char* get_string_to_send (
    float x,
    float y ) [static]
```

Definition at line 63 of file eject.c.

```
64 {
65     char *string_to_send = malloc(sizeof(char) * 10);
66
67     if (!string_to_send)
68         server_err("Malloc failed in get_string_to_send");
69     sprintf(string_to_send, "eject: %d\n", direction_push(x, y));
70     return string_to_send;
71 }
```

14.133.1.5 push_client()

```
void push_client (
    server_t * server,
    client_t * client,
    float x,
    float y )
```

Definition at line 89 of file eject.c.

```
90 {
91     client_t *tmp = server->client->next;
92     int old_x = client->player->pos_x;
93     int old_y = client->player->pos_y;
94     char *msg = get_string_to_send(x, y);
95
96     while (tmp) {
97         if (tmp->client_id == client->client_id || tmp->type == GRAPHICAL) {
```

```

98         tmp = tmp->next;
99         continue;
100    }
101    if (tmp->player->pos_x == old_x && tmp->player->pos_y == old_y)
102        push_single_client(server, client, tmp, msg);
103    tmp = tmp->next;
104 }
105 free(msg);
106 }
```

14.133.1.6 push_eggs()

```

static void push_eggs (
    server_t * server,
    int old_x,
    int old_y ) [static]
```

Definition at line 108 of file eject.c.

```

109 {
110     egg_t *tmp_egg = server->eggs;
111     egg_t *next_egg;
112
113     tmp_egg = server->eggs;
114     while (tmp_egg) {
115         next_egg = tmp_egg->next;
116         if (tmp_egg->pos_x == old_x && tmp_egg->pos_y == old_y) {
117             send_edi_command(server, tmp_egg->egg_id);
118             remove_egg(server, tmp_egg->egg_id, &server->map[old_y][old_x]);
119         }
120         tmp_egg = next_egg;
121     }
122 }
```

14.133.1.7 push_single_client()

```

static void push_single_client (
    server_t * server,
    client_t * client,
    client_t * tmp,
    char * msg ) [static]
```

Definition at line 73 of file eject.c.

```

75 {
76     int new_x = client->player->pos_x +
77         (int)(tmp->player->pos_x == client->player->pos_x);
78     int new_y = client->player->pos_y +
79         (int)(tmp->player->pos_y == client->player->pos_y);
80
81     wrap_position(server, &new_x, &new_y);
82     tmp->player->pos_x = new_x;
83     tmp->player->pos_y = new_y;
84     tile_add_player(&server->map[new_y][new_x], tmp->client_id);
85     send_ppo_command(server, tmp->client_id);
86     write_command_output(tmp->client_fd, msg);
87 }
```

14.133.1.8 wrap_position()

```

static void wrap_position (
    server_t * server,
    int * x,
    int * y ) [static]
```

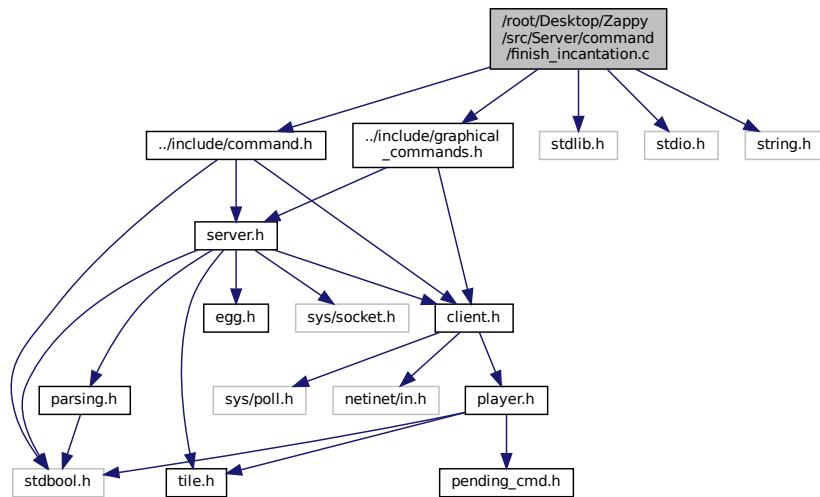
Definition at line 38 of file eject.c.

```

39 {
40     if (*y >= server->parsed_info->height)
41         *y = 0;
42     if (*y < 0)
43         *y = server->parsed_info->height - 1;
44     if (*x < 0)
45         *x = server->parsed_info->width - 1;
46     if (*x >= server->parsed_info->width)
47         *x = 0;
48 }
```

14.134 /root/Desktop/Zappy/src/Server/command/finish_incantation.c File Reference

```
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
Include dependency graph for finish_incantation.c:
```



Functions

- static void handle_incantation_failure (client_t *client)
- static bool remove_resources (tile_t *tile, int level)
- static void handle_win (char *winner, server_t *server)
- static void check_win (client_t *client, server_t *server)
- static void handle_incantation_success (client_t *client, tile_t *tile, int old_level, server_t *server)
- void finish_incantation (server_t *server, client_t *client)

Finalise une incantation en cours.

14.134.1 Function Documentation

14.134.1.1 check_win()

```
static void check_win (
    client_t * client,
    server_t * server ) [static]
```

Definition at line 48 of file finish_incantation.c.

```
49 {
50     int count = 0;
51     char *team_name = client->player->team_name;
52     client_t *temp = server->client;
53
54     while (temp) {
55         if (temp->player && temp->player->level == 8 &&
56             strcmp(temp->player->team_name, team_name) == 0)
57             count++;
58         temp = temp->next;
```

```

59      }
60      if (count >= 6)
61          handle_win(team_name, server);
62 }

```

14.134.1.2 finish_incantation()

```

void finish_incantation (
    server_t * server,
    client_t * client )

```

Finalise une incantation en cours.

Parameters

<i>server</i>	Pointeur vers la structure du serveur
<i>client</i>	Client effectuant l'incantation

Definition at line 81 of file finish_incantation.c.

```

82 {
83     tile_t *tile = &server->map[client->player->pos_y][client->player->pos_x];
84     int old_level;
85
86     if (!client || !client->player || !client->player->is_in_incantation ||
87         client->player->busy_until > server->current_tick)
88         return;
89     old_level = client->player->level;
90     client->player->is_in_incantation = false;
91     if (!can_start_incantation(server, client)) {
92         command_pie(server, client->player->pos_x,
93                     client->player->pos_y, 0);
94         return handle_incantation_failure(client);
95     }
96     handle_incantation_success(client, tile, old_level, server);
97     command_pie(server, client->player->pos_x,
98                 client->player->pos_y, 1);
99     send_plv_to_all(server, client);
100 }

```

14.134.1.3 handle_incantation_failure()

```

static void handle_incantation_failure (
    client_t * client ) [static]

```

Definition at line 14 of file finish_incantation.c.

```

15 {
16     client->player->incantation_leader_id = -1;
17     write_command_output(client->client_fd, "ko\n");
18 }

```

14.134.1.4 handle_incantation_success()

```

static void handle_incantation_success (
    client_t * client,
    tile_t * tile,
    int old_level,
    server_t * server ) [static]

```

Definition at line 64 of file finish_incantation.c.

```

66 {
67     char *level_str;
68
69     client->player->level++;
70     level_str = malloc(20 * sizeof(char));
71     if (client->player->incantation_leader_id == client->client_id)
72         remove_resources(tile, old_level + 1);
73     client->player->incantation_leader_id = -1;
74     sprintf(level_str, "Current level: %d\n", client->player->level);
75     write_command_output(client->client_fd, level_str);
76     free(level_str);
77     if (client->player->level == 8)
78         check_win(client, server);

```

```
79 }
```

14.134.1.5 handle_win()

```
static void handle_win (
    char * winner,
    server_t * server ) [static]
```

Definition at line 41 of file finish_incantation.c.

```
42 {
43     server->should_run = false;
44     command_seg(server, winner);
45     printf("%s team are the winner of this Zappy tournament\n", winner);
46 }
```

14.134.1.6 remove_resources()

```
static bool remove_resources (
    tile_t * tile,
    int level ) [static]
```

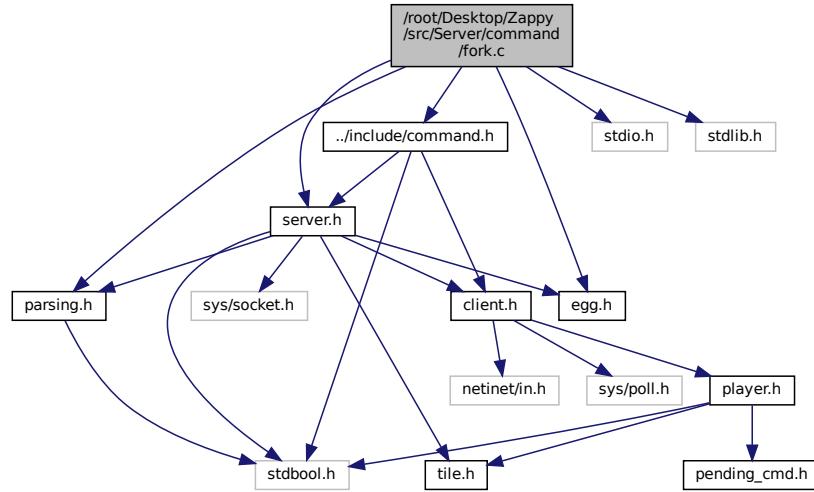
Definition at line 20 of file finish_incantation.c.

```
21 {
22     int required_resources[7][COUNT] = {
23         {1, 0, 0, 0, 0, 0, 0},
24         {1, 1, 0, 0, 0, 0, 0},
25         {2, 1, 1, 0, 0, 0, 0},
26         {2, 2, 1, 1, 0, 0, 0},
27         {4, 1, 1, 2, 1, 0, 0},
28         {4, 3, 2, 1, 1, 1, 0},
29         {6, 1, 2, 3, 2, 2, 1}
30     };
31
32     for (int i = FOOD; i < COUNT; i++) {
33         tile->resources[i] -= required_resources[level - 1][i];
34         if (tile->resources[i] < 0)
35             return false;
36     }
37     return true;
38 }
```

14.135 /root/Desktop/Zappy/src/Server/command/fork.c File Reference

```
#include "../include/command.h"
#include "../include/server.h"
#include "../include/egg.h"
#include "../include/parsing.h"
#include <stdio.h>
#include <stdlib.h>
```

Include dependency graph for fork.c:



Functions

- static int `get_next_egg_id (server_t *server)`
- void `fork_c (server_t *server, client_t *client, char **buffer)`

14.135.1 Function Documentation

14.135.1.1 `fork_c()`

```
void fork_c (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 27 of file fork.c.

```
28 {
29     egg_t *new_egg;
30     int egg_id;
31
32     if (!client || !client->player || arr_len(buffer) != 1) {
33         write_command_output(client->client_fd, "ko\n");
34         return;
35     }
36     egg_id = get_next_egg_id(server);
37     new_egg = create_egg(egg_id, client->player->pos_x, client->player->pos_y,
38                          client->player->team_name);
39     if (!new_egg) {
40         write_command_output(client->client_fd, "ko\n");
41         return;
42     }
43     add_egg(server, new_egg);
44     send_env_command(server, client, egg_id);
45     write_command_output(client->client_fd, "ok\n");
46 }
```

14.135.1.2 `get_next_egg_id()`

```
static int get_next_egg_id (
    server_t * server ) [static]
```

Definition at line 14 of file fork.c.

```

15 {
16     egg_t *current = server->eggs;
17     int max_id = 0;
18
19     while (current) {
20         if (current->egg_id > max_id)
21             max_id = current->egg_id;
22         current = current->next;
23     }
24     return max_id + 1;
25 }

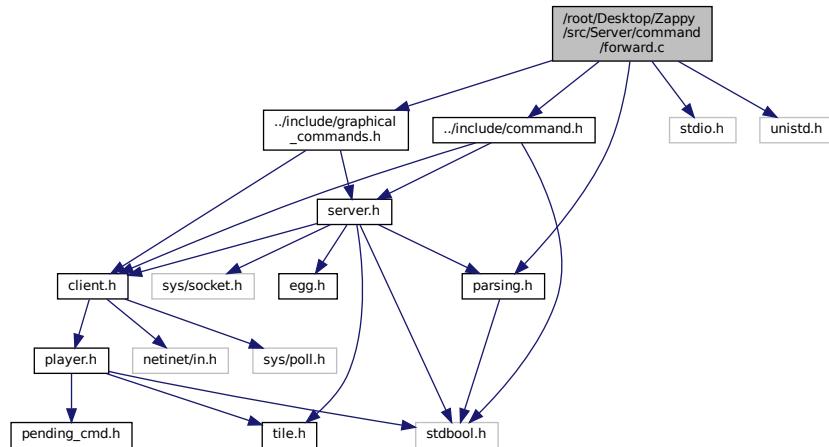
```

14.136 /root/Desktop/Zappy/src/Server/command/forward.c File Reference

```

#include "../include/command.h"
#include "../include/graphical_commands.h"
#include "../include/parsing.h"
#include <stdio.h>
#include <unistd.h>
Include dependency graph for forward.c:

```



Functions

- static void wrap_position (server_t *server, client_t *client)
- static void change_map_pos (server_t *server, client_t *client)
- void forward (server_t *server, client_t *client, char **buffer)

14.136.1 Function Documentation

14.136.1.1 change_map_pos()

```

static void change_map_pos (
    server_t * server,
    client_t * client ) [static]

```

Definition at line 25 of file forward.c.

```

26 {
27     tile_remove_player(
28         &server->map[client->player->pos_y][client->player->pos_x],
29         client->client_id);
30     if (client->player->rotation == UP)

```

```

31     client->player->pos_y--;
32     if (client->player->rotation == DOWN)
33         client->player->pos_y++;
34     if (client->player->rotation == LEFT)
35         client->player->pos_x--;
36     if (client->player->rotation == RIGHT)
37         client->player->pos_x++;
38     wrap_position(server, client);
39     tile_add_player(
40         &server->map[client->player->pos_y][client->player->pos_x],
41         client->client_id);
42 }

```

14.136.1.2 forward()

```

void forward (
    server_t * server,
    client_t * client,
    char ** buffer )

```

Definition at line 44 of file forward.c.

```

45 {
46     if (!client || !client->player || arr_len(buffer) != 1) {
47         write_command_output(client->client_fd, "ko\n");
48         return;
49     }
50     if (client->player->rotation != UP && client->player->rotation != DOWN
51         && client->player->rotation != LEFT
52         && client->player->rotation != RIGHT) {
53         perror("Unexpected forward rotation");
54         write_command_output(client->client_fd, "ko\n");
55         return;
56     }
57     change_map_pos(server, client);
58     send_ppo_command(server, client->client_id);
59     write_command_output(client->client_fd, "ok\n");
60 }

```

14.136.1.3 wrap_position()

```

static void wrap_position (
    server_t * server,
    client_t * client ) [static]

```

Definition at line 13 of file forward.c.

```

14 {
15     if (client->player->pos_y >= server->parsed_info->height)
16         client->player->pos_y = 0;
17     if (client->player->pos_y < 0)
18         client->player->pos_y = server->parsed_info->height - 1;
19     if (client->player->pos_x < 0)
20         client->player->pos_x = server->parsed_info->width - 1;
21     if (client->player->pos_x >= server->parsed_info->width)
22         client->player->pos_x = 0;
23 }

```

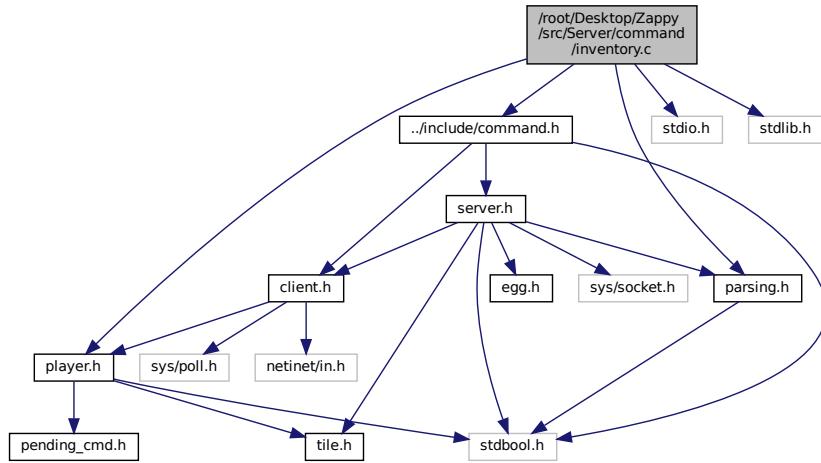
14.137 /root/Desktop/Zappy/src/Server/command/inventory.c File Reference

```

#include "../include/command.h"
#include "../include/player.h"
#include "../include/parsing.h"
#include <stdio.h>
#include <stdlib.h>

```

Include dependency graph for inventory.c:



Functions

- void `inventory (server_t *server, client_t *client, char **buffer)`

14.137.1 Function Documentation

14.137.1.1 inventory()

```
void inventory (
    server_t * server,
    client_t * client,
    char ** buffer )
```

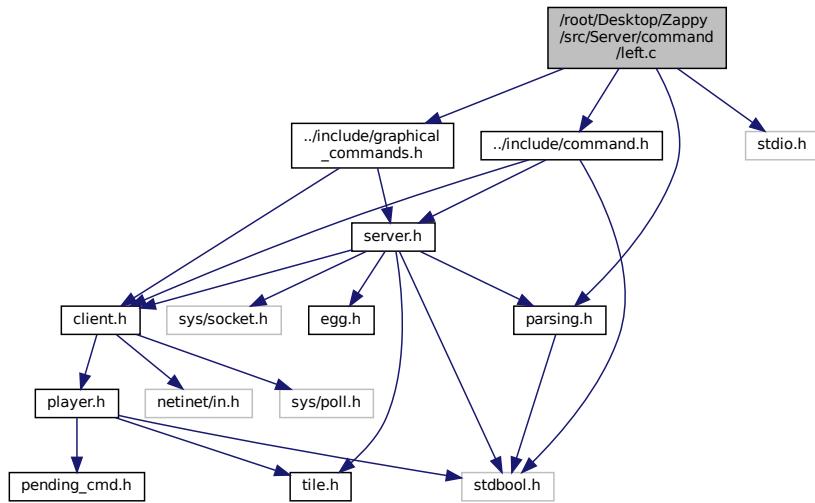
Definition at line 14 of file inventory.c.

```
15 {
16     char *content;
17
18     if (!server || !client || !client->player || arr_len(buffer) != 1)
19         return write_command_output(client->client_fd, "ko\n");
20     content = get_inventory_content(client->player);
21     write_command_output(client->client_fd, content);
22     free(content);
23 }
```

14.138 /root/Desktop/Zappy/src/Server/command/left.c File Reference

```
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include "../include/parsing.h"
#include <stdio.h>
```

Include dependency graph for left.c:



Functions

- static void `change_rot (client_t *client)`
- void `left (server_t *server, client_t *client, char **buffer)`

14.138.1 Function Documentation

14.138.1.1 `change_rot()`

```
static void change_rot (
    client_t * client ) [static]
```

Definition at line 12 of file `left.c`.

```
13 {
14     switch (client->player->rotation) {
15     case UP:
16         client->player->rotation = LEFT;
17         break;
18     case DOWN:
19         client->player->rotation = RIGHT;
20         break;
21     case LEFT:
22         client->player->rotation = DOWN;
23         break;
24     case RIGHT:
25         client->player->rotation = UP;
26         break;
27     default:
28         break;
29 }
30 }
```

14.138.1.2 `left()`

```
void left (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 32 of file `left.c`.

```
33 {
```

```

34     if (!server || !client || !client->player || arr_len(buffer) != 1)
35         return write_command_output(client->client_fd, "ko\n");
36     if (client->player->rotation != UP && client->player->rotation != DOWN
37         && client->player->rotation != LEFT
38         && client->player->rotation != RIGHT) {
39         perror("Unexpected left rotation");
40         return write_command_output(client->client_fd, "ko\n");
41     }
42     change_rot(client);
43     send_ppo_command(server, client->client_id);
44     write_command_output(client->client_fd, "ok\n");
45 }

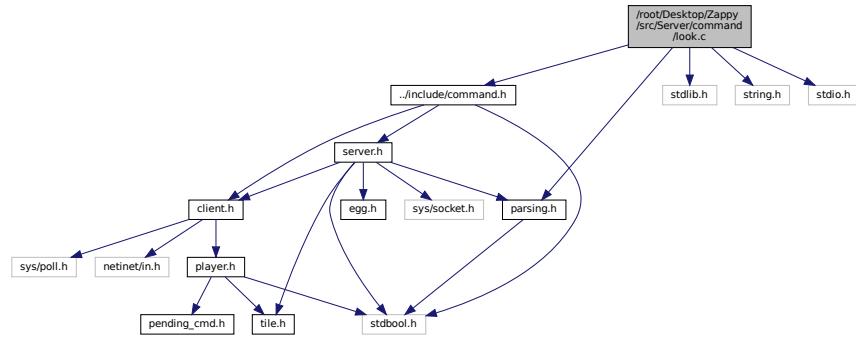
```

14.139 /root/Desktop/Zappy/src/Server/command/look.c File Reference

```

#include "../include/command.h"
#include "../include/parsing.h"
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
Include dependency graph for look.c:

```



Functions

- static int **get_total_size_tile** (int i, char ***tiles, **client_t** *user, **server_t** *server)
- static char * **format_tile** (size_t total_len, char **tiles, int num_tiles)
- static char * **look_tiles** (**client_t** *user, **server_t** *server, int i)
- static int **get_total_size** (char ***level_tiles, **client_t** *user, **server_t** *server)
- static char * **format_look** (size_t total_len, char **level_tiles, **client_t** *user)
- void **look** (**server_t** *server, **client_t** *user, char **buffer)

14.139.1 Function Documentation

14.139.1.1 **format_look()**

```

static char* format_look (
    size_t total_len,
    char ** level_tiles,
    client_t * user ) [static]

```

Definition at line 88 of file look.c.

```

89 {
90     char *res = malloc(sizeof(char) * total_len);
91
92     if (!res)
93         server_err("Malloc failed for res final look");
94     strcpy(res, "[");

```

```

95     for (int i = 0; i <= user->player->level; i++) {
96         if (i > 0)
97             strncat(res, ", ");
98         strncat(res, level_tiles[i]);
99         free(level_tiles[i]);
100    }
101   strncat(res, "]\n");
102   return res;
103 }
```

14.139.1.2 format_tile()

```
static char* format_tile (
    size_t total_len,
    char ** tiles,
    int num_tiles ) [static]
```

Definition at line 35 of file look.c.

```

36 {
37     char *res = malloc(sizeof(char) * total_len);
38
39     if (!res)
40         server_err("Malloc failed for res formatting");
41     res[0] = '\0';
42     for (int j = 0; j < num_tiles; j++) {
43         if (j > 0)
44             strncat(res, ", ");
45         if (strlen(tiles[j]) > 0 && strchr(tiles[j], ' ') != NULL)
46             strncat(res, " ");
47         strncat(res, tiles[j]);
48         free(tiles[j]);
49     }
50     return res;
51 }
```

14.139.1.3 get_total_size()

```
static int get_total_size (
    char *** level_tiles,
    client_t * user,
    server_t * server ) [static]
```

Definition at line 68 of file look.c.

```

70 {
71     int total = 4;
72     int num_levels = user->player->level + 1;
73
74     *level_tiles = malloc(num_levels * sizeof(char *));
75     if (!*level_tiles)
76         server_err("Malloc failed for tiles allocation");
77     for (int i = 0; i <= user->player->level; i++) {
78         (*level_tiles)[i] = look_tiles(user, server, i);
79         if (!(*level_tiles)[i])
80             (*level_tiles)[i] = strdup("");
81         total += strlen((*level_tiles)[i]);
82         if (i > 0)
83             total += 1;
84     }
85     return total;
86 }
```

14.139.1.4 get_total_size_tile()

```
static int get_total_size_tile (
    int i,
    char *** tiles,
    client_t * user,
    server_t * server ) [static]
```

Definition at line 13 of file look.c.

```

15 {
16     size_t total_len = 1;
17     int num_tiles = 2 * i + 1;
18 }
```

```

19     *tiles = malloc(num_tiles * sizeof(char *));
20     if (!*tiles)
21         server_err("Malloc failed for tiles array");
22     for (int j = 0; j < num_tiles; j++) {
23         (*tiles)[j] = check_rota_tiles(user, server, i, j);
24         if (!(*tiles)[j])
25             (*tiles)[j] = strdup("");
26         total_len += strlen((*tiles)[j]);
27         if (strlen((*tiles)[j]) > 0 && strchr((*tiles)[j], ' ') != NULL)
28             total_len += 1;
29         if (j > 0)
30             total_len += 1;
31     }
32     return total_len;
33 }

```

14.139.1.5 look()

```

void look (
    server_t * server,
    client_t * user,
    char ** buffer )

```

Definition at line 105 of file look.c.

```

106 {
107     char **level_tiles = NULL;
108     size_t total_len = 0;
109     char *res;
110
111     if (!server || !user || !user->player || arr_len(buffer) != 1)
112         return write_command_output(user->client_fd, "ko\n");
113     total_len = get_total_size(&level_tiles, user, server);
114     if (!level_tiles)
115         return write_command_output(user->client_fd, "ko\n");
116     res = format_look(total_len, level_tiles, user);
117     free(level_tiles);
118     write_command_output(user->client_fd, res);
119     free(res);
120 }

```

14.139.1.6 look_tiles()

```

static char* look_tiles (
    client_t * user,
    server_t * server,
    int i ) [static]

```

Definition at line 53 of file look.c.

```

54 {
55     int num_tiles = 2 * i + 1;
56     char **tiles = NULL;
57     size_t total_len;
58     char *res;
59
60     total_len = get_total_size_tile(i, &tiles, user, server);
61     if (!tiles)
62         return strdup("");
63     res = format_tile(total_len, tiles, num_tiles);
64     free(tiles);
65     return res;
66 }

```

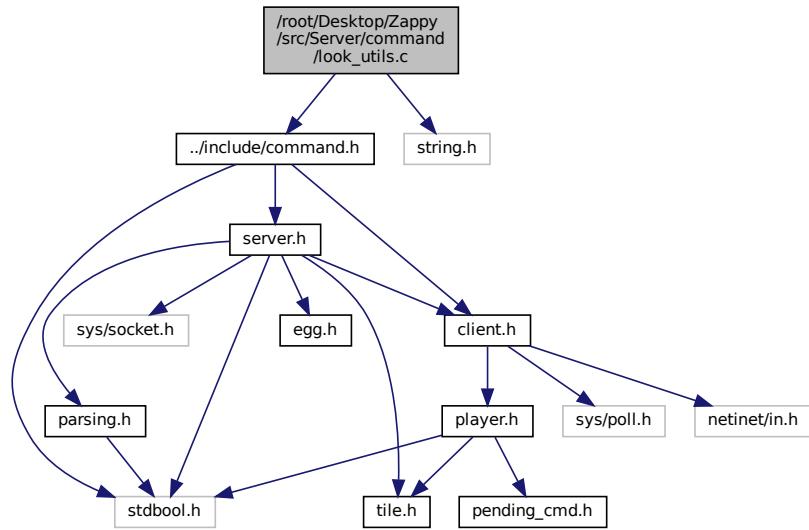
14.140 /root/Desktop/Zappy/src/Server/command/look_utils.c File Reference

```

#include "../include/command.h"
#include <string.h>

```

Include dependency graph for look_utils.c:



Functions

- static char * [check_up \(client_t *user, int i, int j, server_t *server\)](#)
- static char * [check_down \(client_t *user, int i, int j, server_t *server\)](#)
- static char * [check_left \(client_t *user, int i, int j, server_t *server\)](#)
- static char * [check_right \(client_t *user, int i, int j, server_t *server\)](#)
- char * [check_rota_tiles \(client_t *user, server_t *server, int i, int j\)](#)

14.140.1 Function Documentation

14.140.1.1 check_down()

```

static char* check_down (
    client_t * user,
    int i,
    int j,
    server_t * server ) [static]

```

Definition at line 20 of file look_utils.c.

```

21 {
22     int y = (user->player->pos_y + i) % server->parsed_info->height;
23     int x = (user->player->pos_x + i - j + server->parsed_info->width)
24         % server->parsed_info->width;
25
26     return tile_to_str(&server->map[y][x]);
27 }
  
```

14.140.1.2 check_left()

```

static char* check_left (
    client_t * user,
    int i,
    int j,
    server_t * server ) [static]
  
```

Definition at line 29 of file look_utils.c.

```
30 {
31     int y = (user->player->pos_y + j) % server->parsed_info->height;
32     int x = (user->player->pos_x - i + server->parsed_info->width)
33             % server->parsed_info->width;
34
35     return tile_to_str(&server->map[y][x]);
36 }
```

14.140.1.3 check_right()

```
static char* check_right (
    client_t * user,
    int i,
    int j,
    server_t * server ) [static]
```

Definition at line 38 of file look_utils.c.

```
39 {
40     int y = (user->player->pos_y - j + server->parsed_info->height)
41             % server->parsed_info->height;
42     int x = (user->player->pos_x + i) % server->parsed_info->width;
43
44     return tile_to_str(&server->map[y][x]);
45 }
```

14.140.1.4 check_rota_tiles()

```
char* check_rota_tiles (
    client_t * user,
    server_t * server,
    int i,
    int j )
```

Definition at line 47 of file look_utils.c.

```
48 {
49     switch (user->player->rotation) {
50     case UP:
51         return check_up(user, i, j, server);
52     case DOWN:
53         return check_down(user, i, j, server);
54     case LEFT:
55         return check_left(user, i, j, server);
56     case RIGHT:
57         return check_right(user, i, j, server);
58     default:
59         return strdup("");
60     }
61 }
```

14.140.1.5 check_up()

```
static char* check_up (
    client_t * user,
    int i,
    int j,
    server_t * server ) [static]
```

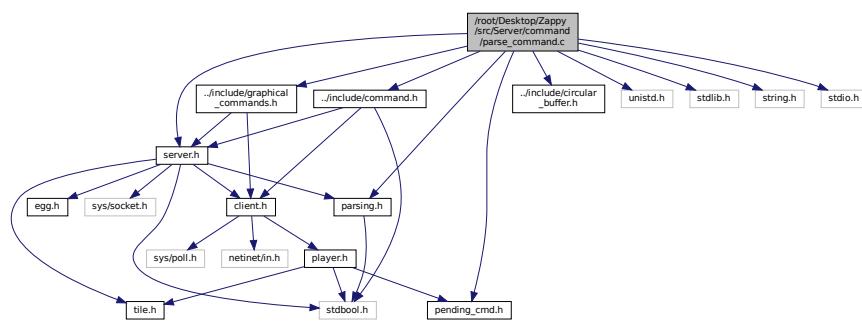
Definition at line 10 of file look_utils.c.

```
11 {
12     int y = (user->player->pos_y - i + server->parsed_info->height)
13             % server->parsed_info->height;
14     int x = (user->player->pos_x - i + j + server->parsed_info->width)
15             % server->parsed_info->width;
16
17     return tile_to_str(&server->map[y][x]);
18 }
```

14.141 /root/Desktop/Zappy/src/Server/command/parse_command.c File Reference

```
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include "../include/server.h"
#include "../include/pending_cmd.h"
#include "../include/circular_buffer.h"
#include "../include/parsing.h"
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
```

Include dependency graph for parse_command.c:



Functions

- static int `check_disconnect` (int bytes_read, `client_t` *user, `server_t` *server)
- `command_data_t get_command_data` (void)
- static bool `execute_graphical_command` (`server_t` *server, `client_t` *user, char *buffer, int cmd_index)
- static bool `execute_if_free` (`server_t` *server, `client_t` *user, char *buffer, int cmd_index)
- static bool `find_and_execute` (`server_t` *server, `client_t` *user, char *buffer)
- static void `send_info_new_client` (`server_t` *server, `client_t` *user)
- void `execute_com` (`server_t` *server, `client_t` *user, char *buffer)
- static void `check_command` (`circular_buffer_t` *temp_buffer, int cmd_length, `server_t` *server, `client_t` *user)
- void `get_message` (`server_t` *server, `client_t` *user)

14.141.1 Function Documentation

14.141.1.1 `check_command()`

```
static void check_command (
    circular_buffer_t * temp_buffer,
    int cmd_length,
    server_t * server,
    client_t * user ) [static]
```

Definition at line 135 of file parse_command.c.

```
137 {
138     char *command;
139
140     command = extract_command(temp_buffer, cmd_length);
141     if (command) {
142         execute_com(server, user, command);
143         free(command);
```

```
144     }
145 }
```

14.141.1.2 check_disconnect()

```
static int check_disconnect (
    int bytes_read,
    client_t * user,
    server_t * server ) [static]
```

Definition at line 18 of file parse_command.c.

```
19 {
20     if (bytes_read <= 0) {
21         if (bytes_read == 0) {
22             cleanup_client(user);
23             remove_fd(server, user->client_poll->fd);
24             return 1;
25         }
26         return 1;
27     }
28     return 0;
29 }
```

14.141.1.3 execute_com()

```
void execute_com (
    server_t * server,
    client_t * user,
    char * buffer )
```

Definition at line 115 of file parse_command.c.

```
116 {
117     if (!user)
118         return;
119     if (!user->is_fully_connected && can_connect(server, user, buffer)) {
120         user->is_fully_connected = true;
121         if (user->type == GRAPHICAL) {
122             add_graphic_client(server, user);
123             send_map_info_to_one_client(server, user);
124             return;
125         } else
126             return send_info_new_client(server, user);
127     }
128     if (!find_and_execute(server, user, buffer)) {
129         if (user->type == GRAPHICAL)
130             return write_command_output(user->client_fd, "suc\n");
131         write_command_output(user->client_fd, "ko\n");
132     }
133 }
```

14.141.1.4 execute_graphical_command()

```
static bool execute_graphical_command (
    server_t * server,
    client_t * user,
    char * buffer,
    int cmd_index ) [static]
```

Definition at line 53 of file parse_command.c.

```
55 {
56     command_data_t data = get_command_data();
57     char **args = str_to_word_arr(buffer, " ");
58
59     data.functions[cmd_index](server, user, args);
60     free_arr(args);
61     return true;
62 }
```

14.141.1.5 execute_if_free()

```
static bool execute_if_free (
```

```
    server_t * server,
    client_t * user,
    char * buffer,
    int cmd_index ) [static]
```

Definition at line 64 of file parse_command.c.

```
66 {
67     if (user->type == GRAPHICAL)
68         return execute_graphical_command(server, user, buffer, cmd_index);
69     if (user->type == AI && user->player->busy_until <= server->current_tick) {
70         add_pending_cmd(user, server, buffer, cmd_index);
71         return true;
72     } else {
73         if (user->player->queue_size < 10) {
74             add_to_command_queue(server, user, buffer);
75             return true;
76         } else
77             return true;
78     }
79 }
```

14.141.1.6 find_and_execute()

```
static bool find_and_execute (
    server_t * server,
    client_t * user,
    char * buffer ) [static]
```

Definition at line 81 of file parse_command.c.

```
82 {
83     command_data_t data = get_command_data();
84
85     for (int i = 0; data.commands[i] != NULL; i++) {
86         if (strncmp(buffer, data.commands[i], strlen(data.commands[i])) == 0 &&
87             user->type == data.accepted_types[i])
88             return execute_if_free(server, user, buffer, i);
89     }
90     return false;
91 }
```

14.141.1.7 get_command_data()

```
command_data_t get_command_data (
    void )
```

Definition at line 31 of file parse_command.c.

```
32 {
33     static const char *comm_char[] = {"Forward", "Right", "Left",
34         "Inventory", "Look", "Eject", "Connect_nbr", "Take", "Set",
35         "Incantation", "Fork", "Broadcast", "msz", "bct", "mtc",
36         "tna", "ppo", "plv", "pin", "sgt", "sst", NULL};
37     static void (*comm_func[]) (server_t *, client_t *, char **) =
38         {forward, right, left, inventory, look, eject,
39         connect_nbr, take_object, set_object, start_incantation,
40         fork_c, broadcast, command_msz, command_bct, command_mtc,
41         command_tna, command_ppo, command_plv, command_pin,
42         command_sgt, command_sst, NULL};
43     static int comm_times[] = {7, 7, 7, 1, 7, 7, 0, 7, 7, 300, 42, 7, 0,
44         0, 0, 0, 0, 0, 0, 0};
45     static enum client_type_e accepted_types[] = {AI, AI, AI, AI, AI,
46         AI, AI, AI, AI, AI, GRAPHICAL, GRAPHICAL, GRAPHICAL,
47         GRAPHICAL, GRAPHICAL, GRAPHICAL, GRAPHICAL, GRAPHICAL};
48     command_data_t data = {comm_char, comm_func, comm_times, accepted_types};
49
50     return data;
51 }
```

14.141.1.8 get_message()

```
void get_message (
    server_t * server,
    client_t * user )
```

Definition at line 147 of file parse_command.c.

```
148 {
```

```

149     circular_buffer_t temp_buffer;
150     char byte;
151     int bytes_read;
152     int cmd_length;
153
154     init_circular_buffer(&temp_buffer);
155     while (1) {
156         bytes_read = read(user->client_poll->fd, &byte, 1);
157         if (check_disconnect(bytes_read, user, server) == 1)
158             return;
159         if (add_to_circular_buffer(&temp_buffer, byte) == -1) {
160             write_command_output(user->client_fd, "ko\n");
161             return;
162         }
163         cmd_length = find_command_end(&temp_buffer);
164         if (cmd_length > 0) {
165             check_command(&temp_buffer, cmd_length, server, user);
166             break;
167         }
168     }
169 }
```

14.141.1.9 send_info_new_client()

```
static void send_info_new_client (
    server_t *server,
    client_t *user ) [static]
```

Definition at line 93 of file parse_command.c.

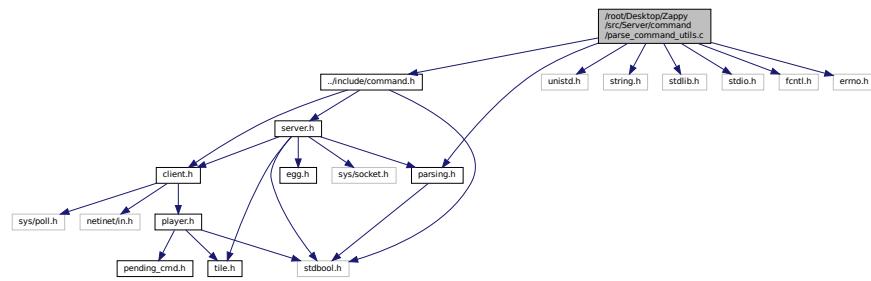
```

94 {
95     char *tmp_string = NULL;
96     int len1 = snprintf(NULL, 0, "%d\n", user->client_id);
97     int len2 = snprintf(NULL, 0, "%d %d\n",
98         server->parsed_info->width,
99         server->parsed_info->height);
100
101    tmp_string = malloc(len1 + 1);
102    sprintf(tmp_string, "%d\n", connect_nbr_srv(server,
103        user->player->team_name));
104    write_command_output(user->client_fd, tmp_string);
105    free(tmp_string);
106    tmp_string = malloc(len2 + 1);
107    sprintf(tmp_string, "%d %d\n",
108        server->parsed_info->width,
109        server->parsed_info->height);
110    write_command_output(user->client_fd, tmp_string);
111    free(tmp_string);
112    send_pnw_command_to_all(server, user);
113 }
```

14.142 /root/Desktop/Zappy/src/Server/command/parse_command_utils.c File Reference

```
#include "../include/command.h"
#include "../include/parsing.h"
#include <unistd.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>
#include <errno.h>
```

Include dependency graph for parse_command_utils.c:



Functions

- void `cleanup_pending (player_t *player)`
- void `add_pending_cmd (client_t *user, server_t *server, char *buffer, int cmd_index)`
- void `cleanup_player_queue (player_t *player)`
- void `cleanup_client (client_t *client)`
- void `write_command_output (int client_fd, char *msg)`

14.142.1 Function Documentation

14.142.1.1 add_pending_cmd()

```
void add_pending_cmd (
    client_t * user,
    server_t * server,
    char * buffer,
    int cmd_index )
```

Definition at line 28 of file parse_command_utils.c.

```
30 {
31     command_data_t data = get_command_data();
32     char **tmp = NULL;
33
34     if (cmd_index == 9) {
35         tmp = str_to_word_arr(buffer, " ");
36         start_incantation(server, user, tmp);
37         return free_arr(tmp);
38     }
39     if (cmd_index == 10)
40         command_pfk(server, user);
41     user->player->pending_cmd->args = strdup(buffer);
42     user->player->pending_cmd->func = data.functions[cmd_index];
43     if (data.times[cmd_index] > 0)
44         user->player->busy_until =
45             server->current_tick + data.times[cmd_index];
46 }
```

14.142.1.2 cleanup_client()

```
void cleanup_client (
    client_t * client )
```

Definition at line 63 of file parse_command_utils.c.

```
64 {
65     if (!client)
66         return;
67     if (client->player) {
68         cleanup_player_queue(client->player);
69         cleanup_pending(client->player);
70         free(client->player);
71         client->player = NULL;
```

```
72     }
73 }
```

14.142.1.3 cleanup_pending()

```
void cleanup_pending (
    player_t * player )
```

Definition at line 16 of file parse_command_utils.c.

```
17 {
18     if (!player || !player->pending_cmd)
19         return;
20     if (player->pending_cmd->args) {
21         free(player->pending_cmd->args);
22         player->pending_cmd->args = NULL;
23     }
24     free(player->pending_cmd);
25     player->pending_cmd = NULL;
26 }
```

14.142.1.4 cleanup_player_queue()

```
void cleanup_player_queue (
    player_t * player )
```

Definition at line 48 of file parse_command_utils.c.

```
49 {
50     if (!player || !player->command_queue)
51         return;
52     for (int i = 0; i < player->queue_size; i++) {
53         if (player->command_queue[i]) {
54             free(player->command_queue[i]);
55             player->command_queue[i] = NULL;
56         }
57     }
58     free(player->command_queue);
59     player->command_queue = NULL;
60     player->queue_size = 0;
61 }
```

14.142.1.5 write_command_output()

```
void write_command_output (
    int client_fd,
    char * msg )
```

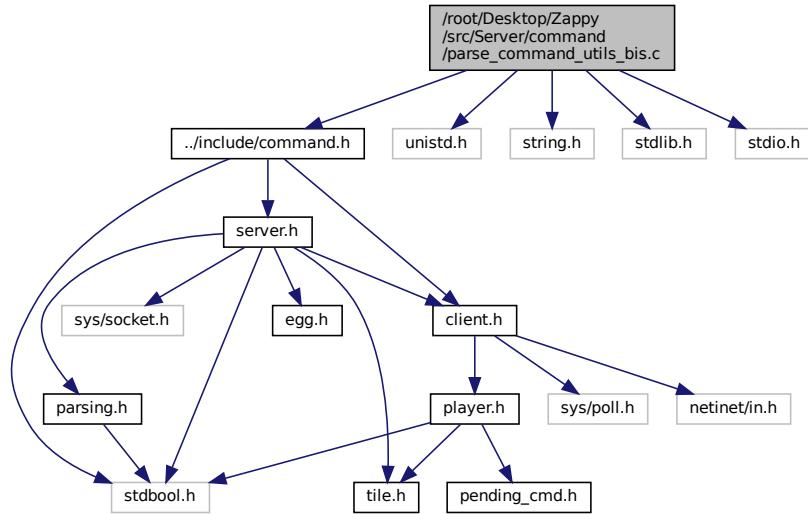
Definition at line 75 of file parse_command_utils.c.

```
76 {
77     if (fcntl(client_fd, F_GETFD) == -1) {
78         perror("FD isn't up anymore\n");
79     } else {
80         write(client_fd, msg, strlen(msg));
81     }
82 }
```

14.143 /root/Desktop/Zappy/src/Server/command/parse_command_utils_bis.c File Reference

```
#include "../include/command.h"
#include <unistd.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
```

Include dependency graph for parse_command_utils_bis.c:



Functions

- static bool **is_valid_team_name** (char *team_name, **server_t** *server, **client_t** *user)
- bool **can_connect** (**server_t** *server, **client_t** *user, char *buffer)

14.143.1 Function Documentation

14.143.1.1 can_connect()

```
bool can_connect (
    server_t * server,
    client_t * user,
    char * buffer )
```

Definition at line 38 of file parse_command_utils_bis.c.

```
39 {
40     if (!is_valid_team_name(buffer, server, user)){
41         write_command_output(user->client_fd, "ko\n");
42         return false;
43     }
44     if (user->type != GRAPHICAL &&
45         connect_nbr_srv(server, user->player->team_name) < 0) {
46         if (user->player->team_name) {
47             free(user->player->team_name);
48             user->player->team_name = NULL;
49         }
50         write_command_output(user->client_fd, "ko\n");
51         return false;
52     }
53     return true;
54 }
```

14.143.1.2 is_valid_team_name()

```
static bool is_valid_team_name (
    char * team_name,
    server_t * server,
    client_t * user ) [static]
```

Definition at line 13 of file parse_command_utils_bis.c.

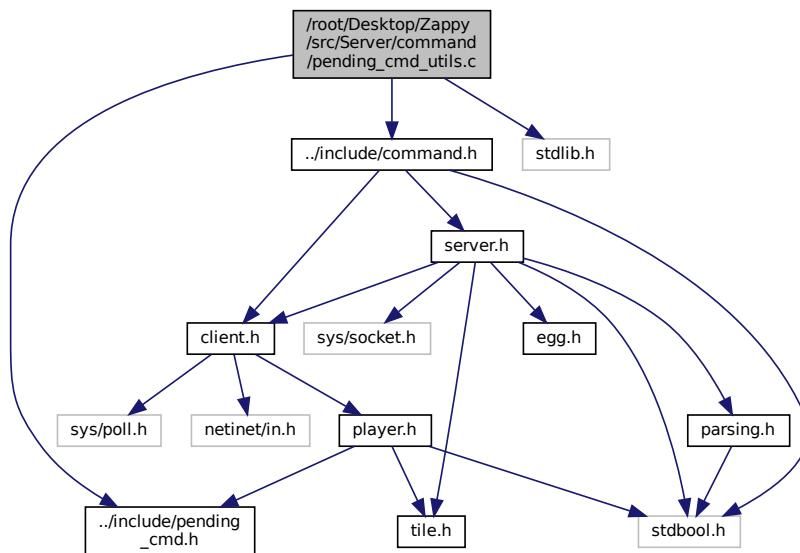
```

15 {
16     if (!team_name || !server ||
17         !server->parsed_info || !server->parsed_info->names)
18     return false;
19     if (strlen(team_name) < 2 || team_name[strlen(team_name) - 1] != '\n')
20     return false;
21     team_name[strlen(team_name) - 1] = '\0';
22     if (strcmp(team_name, "GRAPHIC") == 0) {
23         user->type = GRAPHICAL;
24         user->player = NULL;
25         return true;
26     }
27     for (int i = 0; server->parsed_info->names[i] != NULL; i++) {
28         if (strcmp(team_name, server->parsed_info->names[i]) == 0) {
29             user->player->team_name = strdup(team_name);
30             user->type = AI;
31             init_new_player_pos(server, user);
32             return true;
33         }
34     }
35     return false;
36 }
```

14.144 /root/Desktop/Zappy/src/Server/command/pending_cmd_utils.c File Reference

```
#include "../include/pending_cmd.h"
#include "../include/command.h"
#include <stdlib.h>
```

Include dependency graph for pending_cmd_utils.c:



Functions

- void execute_pending_cmd (server_t *server, client_t *client)

14.144.1 Function Documentation

14.144.1.1 execute_pending_cmd()

```
void execute_pending_cmd (
    server_t * server,
    client_t * client )
```

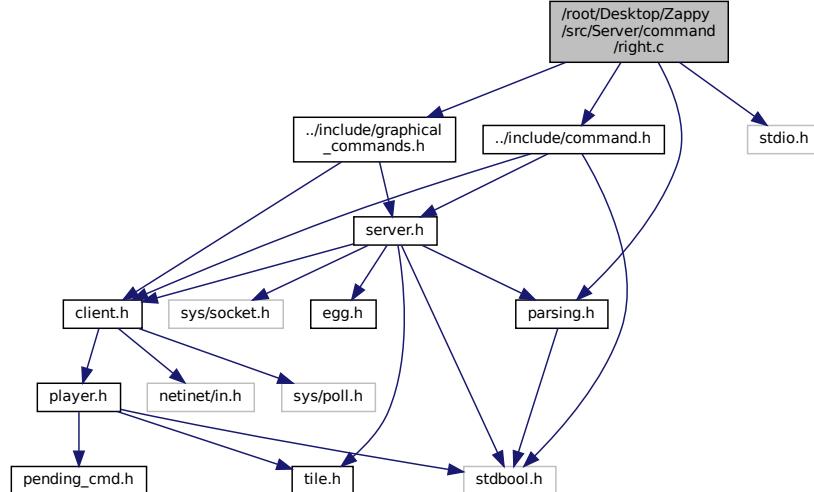
Definition at line 11 of file pending_cmd_utils.c.

```
12 {
13     pending_cmd_t *cmd;
14     char **args_array;
15
16     if (!client || !client->player || !client->player->pending_cmd)
17         return;
18     cmd = client->player->pending_cmd;
19     if (cmd->func && cmd->args) {
20         args_array = str_to_word_arr(cmd->args, " ");
21         cmd->func(server, client, args_array);
22         free_arr(args_array);
23     }
24     if (cmd->args)
25         free(cmd->args);
26     cmd->args = NULL;
27     cmd->func = NULL;
28 }
```

14.145 /root/Desktop/Zappy/src/Server/command/right.c File Reference

```
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include "../include/parsing.h"
#include <stdio.h>
```

Include dependency graph for right.c:



Functions

- static void [change_rot \(client_t *client\)](#)
- void [right \(server_t *server, client_t *client, char **buffer\)](#)

14.145.1 Function Documentation

14.145.1.1 change_rot()

```
static void change_rot (
    client_t * client ) [static]
```

Definition at line 12 of file right.c.

```
13 {
14     switch (client->player->rotation) {
15     case UP:
16         client->player->rotation = RIGHT;
17         break;
18     case DOWN:
19         client->player->rotation = LEFT;
20         break;
21     case LEFT:
22         client->player->rotation = UP;
23         break;
24     case RIGHT:
25         client->player->rotation = DOWN;
26         break;
27     default:
28         break;
29 }
30 }
```

14.145.1.2 right()

```
void right (
    server_t * server,
    client_t * client,
    char ** buffer )
```

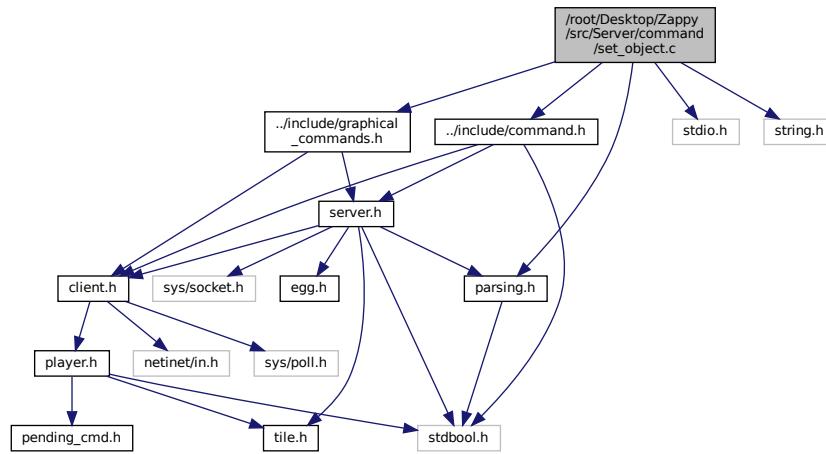
Definition at line 32 of file right.c.

```
33 {
34     if (!server || !client || !client->player || arr_len(buffer) != 1)
35         return write_command_output(client->client_fd, "ko\n");
36     if (client->player->rotation != UP && client->player->rotation != DOWN
37         && client->player->rotation != LEFT
38         && client->player->rotation != RIGHT) {
39         perror("Unexpected right rotation");
40         return write_command_output(client->client_fd, "ko\n");
41     }
42     change_rot(client);
43     send_ppo_command(server, client->client_id);
44     write_command_output(client->client_fd, "ok\n");
45 }
```

14.146 /root/Desktop/Zappy/src/Server/command/set_object.c File Reference

```
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include "../include/parsing.h"
#include <stdio.h>
#include <string.h>
```

Include dependency graph for set_object.c:



Functions

- void [set_object \(server_t *server, client_t *client, char **buffer\)](#)

14.146.1 Function Documentation

14.146.1.1 set_object()

```
void set_object (
    server_t * server,
    client_t * client,
    char ** buffer )
```

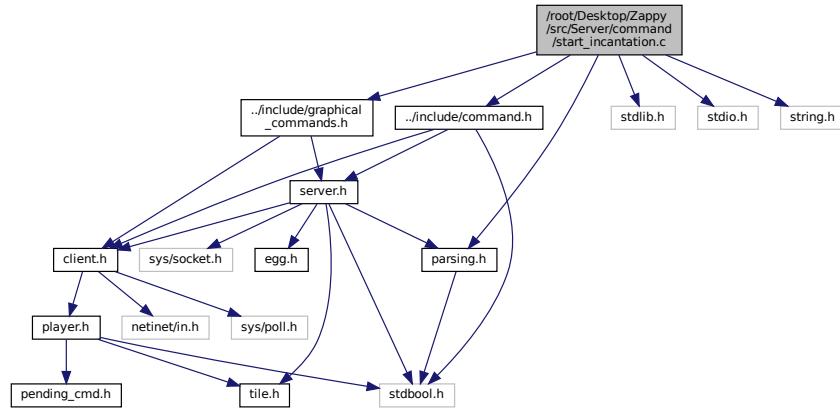
Definition at line 14 of file [set_object.c](#).

```
15 {
16     resource_type_t resource_type;
17
18     if (!client || !client->player || arr_len(buffer) != 2)
19         return write_command_output(client->client_fd, "ko\n");
20     resource_type = determine_type(buffer[1]);
21     if (resource_type == COUNT)
22         return write_command_output(client->client_fd, "ko\n");
23     if (how_many_in_inventory(client->player, resource_type) <= 0)
24         return write_command_output(client->client_fd, "ko\n");
25     remove_item_from_inventory(client->player, resource_type, 1);
26     server->map[client->player->pos_y]
27         [client->player->pos_x].resources[resource_type]++;
28     server->current_resources[resource_type]++;
29     command_pdr(server, client, resource_type);
30     send_pin_to_all(server, client);
31     send_bct_to_all_graphical_clients(server, client->player->pos_x,
32         client->player->pos_y);
33     write_command_output(client->client_fd, "ok\n");
34 }
```

14.147 /root/Desktop/Zappy/src/Server/command/start_incantation.c File Reference

```
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include "../include/parsing.h"
#include <stdlib.h>
```

```
#include <stdio.h>
#include <string.h>
Include dependency graph for start_incantation.c:
```



Functions

- static `client_t * get_client_by_id_player (server_t *server, int player_id)`
- static `int nb_valid_players_on_tile (server_t *server, tile_t *tile, int current_level)`
- static `void set_busy_until (server_t *server, client_t *client, int duration)`
- static `void set_busy_all (server_t *server, client_t *client, int duration)`
- static `bool has_enough_resources (tile_t *tile, int level)`
- static `int how_many_players_needed (int level)`
- `bool can_start_incantation (server_t *server, client_t *client)`
- `void start_incantation (server_t *server, client_t *client, char **buffer)`

14.147.1 Function Documentation

14.147.1.1 can_start_incantation()

```
bool can_start_incantation (
    server_t * server,
    client_t * client )
```

Definition at line 91 of file start_incantation.c.

```

92 {
93     int required_players = 0;
94     int prerequisite_level = client->player->level + 1;
95     tile_t *tile = &server->map[client->player->pos_y][client->player->pos_x];
96     int current_players = 0;
97
98     if (prerequisite_level > 7 ||
99         !has_enough_resources(tile, prerequisite_level))
100         return false;
101     required_players = how_many_players_needed(prerequisite_level);
102     current_players =
103         nb_valid_players_on_tile(server, tile, client->player->level);
104     if (current_players < required_players)
105         return false;
106     return true;
107 }
```

14.147.1.2 get_client_by_id_player()

```
static client_t* get_client_by_id_player (
    server_t * server,
    int player_id ) [static]

Definition at line 15 of file start_incantation.c.
16 {
17     client_t *tmp = server->client->next;
18
19     while (tmp != NULL) {
20         if (tmp->player != NULL && tmp->client_id == player_id)
21             return tmp;
22         tmp = tmp->next;
23     }
24     return NULL;
25 }
```

14.147.1.3 has_enough_resources()

```
static bool has_enough_resources (
    tile_t * tile,
    int level ) [static]
```

Definition at line 63 of file start_incantation.c.

```
64 {
65     int required_resources[7][COUNT] = {
66         {1, 0, 0, 0, 0, 0, 0},
67         {1, 1, 0, 0, 0, 0, 0},
68         {2, 1, 1, 0, 0, 0, 0},
69         {2, 2, 1, 1, 0, 0, 0},
70         {4, 1, 1, 2, 1, 0, 0},
71         {4, 3, 2, 1, 1, 1, 0},
72         {6, 1, 2, 3, 2, 2, 1}
73     };
74
75     for (int i = FOOD; i < COUNT; i++) {
76         if (tile->resources[i] < required_resources[level - 1][i])
77             return false;
78     }
79     return true;
80 }
```

14.147.1.4 how_many_players_needed()

```
static int how_many_players_needed (
    int level ) [static]
```

Definition at line 82 of file start_incantation.c.

```
83 {
84     int required_players[7] = {1, 2, 2, 4, 4, 6, 6};
85
86     if (level < 1 || level > 7)
87         return -1;
88     return required_players[level - 1];
89 }
```

14.147.1.5 nb_valid_players_on_tile()

```
static int nb_valid_players_on_tile (
    server_t * server,
    tile_t * tile,
    int current_level ) [static]
```

Definition at line 27 of file start_incantation.c.

```
29 {
30     int count = 0;
31
32     for (int i = 0; i < tile->player_count; i++) {
33         if (tile->player_ids[i] != -1 &&
34             get_client_by_id_player(server, tile->player_ids[i])->
35             player->level == current_level) {
36             count++;
37         }
38     }
```

```
39     return count;
40 }
```

14.147.1.6 set_busy_all()

```
static void set_busy_all (
    server_t * server,
    client_t * client,
    int duration ) [static]
```

Definition at line 48 of file start_incantation.c.

```
49 {
50     tile_t *tile = &server->map[client->player->pos_y][client->player->pos_x];
51     client_t *tmp_client;
52
53     for (int i = 0; i < tile->player_count; i++) {
54         tmp_client = get_client_by_id_player(server, tile->player_ids[i]);
55         if (tmp_client && tmp_client->player &&
56             tmp_client->player->level == client->player->level) {
57             tmp_client->player->incantation_leader_id = client->client_id;
58             set_busy_until(server, tmp_client, duration);
59         }
60     }
61 }
```

14.147.1.7 set_busy_until()

```
static void set_busy_until (
    server_t * server,
    client_t * client,
    int duration ) [static]
```

Definition at line 42 of file start_incantation.c.

```
43 {
44     client->player->busy_until = server->current_tick + duration;
45     client->player->is_in_incantation = true;
46 }
```

14.147.1.8 start_incantation()

```
void start_incantation (
    server_t * server,
    client_t * client,
    char ** buffer )
```

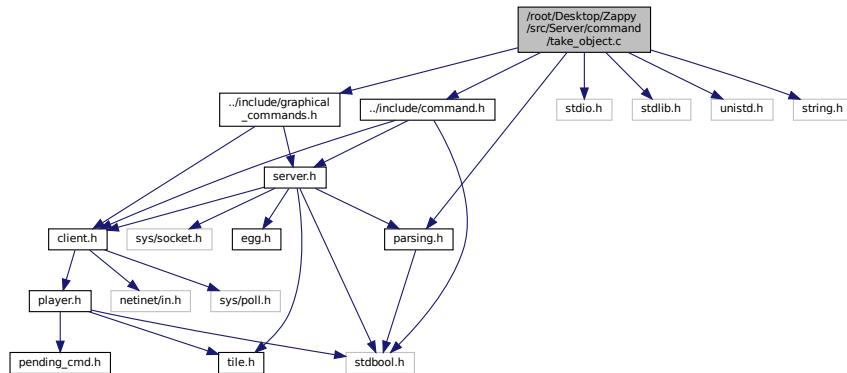
Definition at line 109 of file start_incantation.c.

```
110 {
111     if (!server || !client || !client->player || arr_len(buffer) != 1)
112         return write_command_output(client->client_fd, "ko\n");
113     if (!can_start_incantation(server, client))
114         return write_command_output(client->client_fd, "ko\n");
115     command_pic(server, client->player->pos_x, client->player->pos_y,
116                 client->player->level);
117     set_busy_all(server, client, 300);
118     client->player->is_in_incantation = true;
119     client->player->incantation_leader_id = client->client_id;
120     write_command_output(client->client_fd, "Elevation underway\n");
121 }
```

14.148 /root/Desktop/Zappy/src/Server/command/take_object.c File Reference

```
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include "../include/parsing.h"
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
```

```
#include <string.h>
Include dependency graph for take_object.c:
```



Functions

- `resource_type_t determine_type (char *resource_string)`
- `static void update_resources (server_t *server, client_t *client, resource_type_t type)`
- `void take_object (server_t *server, client_t *client, char **buffer)`

14.148.1 Function Documentation

14.148.1.1 `determine_type()`

```
resource_type_t determine_type (
    char * resource_string )
```

Definition at line 16 of file `take_object.c`.

```
17 {
18     if (strncmp(resource_string, "food", 4) == 0)
19         return FOOD;
20     if (strncmp(resource_string, "linemate", 8) == 0)
21         return LINEMATE;
22     if (strncmp(resource_string, "deraumere", 9) == 0)
23         return DERAUMERE;
24     if (strncmp(resource_string, "sibur", 5) == 0)
25         return SIBUR;
26     if (strncmp(resource_string, "mendiane", 8) == 0)
27         return MENDIANE;
28     if (strncmp(resource_string, "phiras", 6) == 0)
29         return PHIRAS;
30     if (strncmp(resource_string, "thystame", 8) == 0)
31         return THYSTAME;
32     return COUNT;
33 }
```

14.148.1.2 `take_object()`

```
void take_object (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 43 of file `take_object.c`.

```
44 {
45     resource_type_t type;
46
47     if (!server || !client || !client->player || arr_len(buffer) != 2)
48         return write_command_output(client->client_fd, "ko\n");
49     type = determine_type(buffer[1]);
```

```

50     if (type == COUNT)
51         return write_command_output(client->client_fd, "ko\n");
52     if (server->map[client->player->pos_y]
53         [client->player->pos_x].resources[type] > 0) {
54         update_resources(server, client, type);
55         add_item_to_inventory(client->player, type, 1);
56         command_pgt(server, client, type);
57         send_bct_to_all_graphical_clients(server, client->player->pos_x,
58             client->player->pos_y);
59         send_pin_to_all(server, client);
60         write_command_output(client->client_fd, "ok\n");
61     } else
62         write_command_output(client->client_fd, "ko\n");
63 }

```

14.148.1.3 update_resources()

```

static void update_resources (
    server_t * server,
    client_t * client,
    resource_type_t type ) [static]

```

Definition at line 35 of file take_object.c.

```

37 {
38     server->map[client->player->pos_y]
39     [client->player->pos_x].resources[type]--;
40     server->current_resources[type]--;
41 }

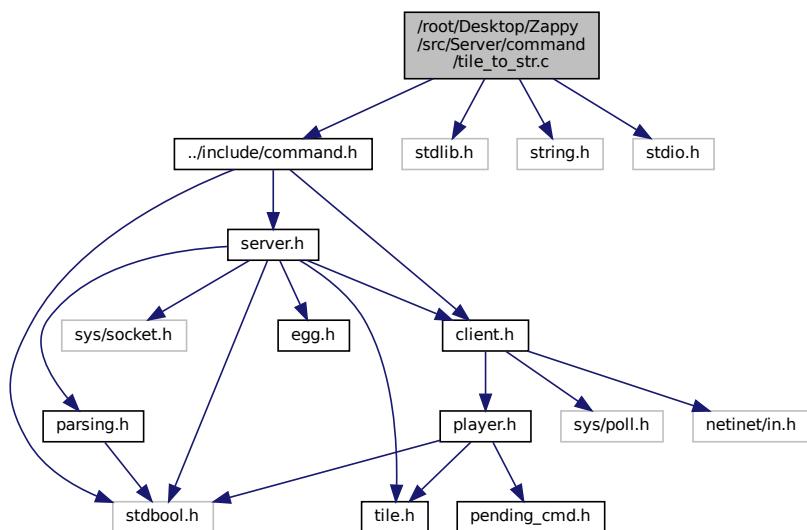
```

14.149 /root/Desktop/Zappy/src/Server/command/tile_to_str.c File Reference

```

#include "../include/command.h"
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
Include dependency graph for tile_to_str.c:

```



Functions

- static char * repeat_word (char *word, int nb)

- static int `calculate_total_size` (int *resources, char **names)
- static char * `format_str_resources` (char *names, int resources, int first_resource, char *res)
- static char * `add_resources` (int *resources)
- char * `tile_to_str` (tile_t *tile)

14.149.1 Function Documentation

14.149.1.1 `add_resources()`

```
static char* add_resources (
    int * resources ) [static]
```

Definition at line 57 of file tile_to_str.c.

```
58 {
59     char *names[COUNT] = {"food", "linemate", "deraumere", "sibur", "mendiane",
60     "phiras", "thystame"};
61     size_t total_size = calculate_total_size(resources, names);
62     char *res = malloc(total_size);
63     int first_resource = 1;
64     char *temp;
65
66     if (!res)
67         server_err("Malloc failed res add_resources");
68     res[0] = '\0';
69     for (int i = 0; i < COUNT; ++i) {
70         if (resources[i] > 0) {
71             temp = format_str_resources(names[i], resources[i],
72                 first_resource, res);
73             strcat(res, temp);
74             free(temp);
75             first_resource = 0;
76         }
77     }
78     return res;
79 }
```

14.149.1.2 `calculate_total_size()`

```
static int calculate_total_size (
    int * resources,
    char ** names ) [static]
```

Definition at line 30 of file tile_to_str.c.

```
31 {
32     int non_empty_count = 0;
33     int total_size = 2;
34
35     for (int i = 0; i < COUNT; ++i) {
36         if (resources[i] > 0) {
37             total_size += (strlen(names[i]) *
38                 resources[i]) + (resources[i] - 1);
39             non_empty_count++;
40         }
41     }
42     if (non_empty_count > 1)
43         total_size += non_empty_count - 1;
44     return total_size;
45 }
```

14.149.1.3 `format_str_resources()`

```
static char* format_str_resources (
    char * names,
    int resources,
    int first_resource,
    char * res ) [static]
```

Definition at line 47 of file tile_to_str.c.

```
49 {
50     char *temp = repeat_word(names, resources);
51 }
```

```

52     if (!first_resource)
53         strcat(res, " ");
54     return temp;
55 }

```

14.149.1.4 repeat_word()

```

static char* repeat_word (
    char * word,
    int nb ) [static]

```

Definition at line 12 of file tile_to_str.c.

```

13 {
14     char *res;
15
16     if (nb <= 0)
17         return strdup("");
18     res = calloc(((strlen(word) * nb) + (nb + 1)), sizeof(char));
19     if (!res)
20         server_err("Malloc failed for res repeat_word");
21     res[0] = '\0';
22     for (int i = 0; i < nb; ++i) {
23         if (i > 0)
24             strcat(res, " ");
25         strcat(res, word);
26     }
27     return res;
28 }

```

14.149.1.5 tile_to_str()

```

char* tile_to_str (
    tile_t * tile )

```

Definition at line 81 of file tile_to_str.c.

```

82 {
83     char *player = repeat_word("player", tile->player_count);
84     char *resources = add_resources(tile->resources);
85     int p_len = strlen(player);
86     int r_len = strlen(resources);
87     char *res = malloc(p_len + r_len + 2);
88
89     if (!res)
90         server_err("Malloc failed res tile_to_str");
91     res[0] = '\0';
92     if (p_len)
93         strcat(res, player);
94     if (p_len && r_len)
95         strcat(res, " ");
96     if (r_len)
97         strcat(res, resources);
98     free(player);
99     free(resources);
100    return res;
101 }

```

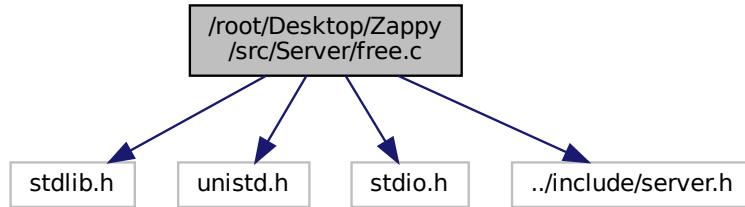
14.150 /root/Desktop/Zappy/src/Server/free.c File Reference

```

#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include "../include/server.h"

```

Include dependency graph for free.c:



Functions

- static void `free_client (server_t *server)`
- static void `free_eggs (server_t *server)`
- static void `free_tiles (server_t *server, int i, parsing_info_t *parsed_info)`
- static void `free_map (server_t *server, parsing_info_t *parsed_info)`
- static void `free_parsed_info (parsing_info_t *p_info)`
- static void `free_poll_mana (server_t *server)`
- void `free_all (server_t *server, parsing_info_t *parsed_info)`
Libère toute la mémoire allouée par le serveur.
- void `free_arr (char **array)`

14.150.1 Function Documentation

14.150.1.1 `free_all()`

```
void free_all (
    server_t * server,
    parsing_info_t * parsed_info )
```

Libère toute la mémoire allouée par le serveur.

Parameters

<code>server</code>	Pointeur vers la structure du serveur
<code>parsed_info</code>	Informations de configuration à libérer

Definition at line 90 of file free.c.

```
91 {
92     if (!server)
93         return;
94     free_client(server);
95     free_eggs(server);
96     free_map(server, parsed_info);
97     if (server->poll_manager)
98         free_poll_mana(server);
99     if (parsed_info)
100         free_parsed_info(parsed_info);
101     if (server->parsed_info) {
102         free_parsed_info(server->parsed_info);
103         free(server->parsed_info);
104         server->parsed_info = NULL;
105     }
106 }
```

14.150.1.2 free_arr()

```
void free_arr (
    char ** array )
Definition at line 108 of file free.c.
109 {
110     if (!array)
111         return;
112     for (int i = 0; array[i] != NULL; i++) {
113         if (array[i] == NULL)
114             break;
115         free(array[i]);
116     }
117     free(array);
118 }
```

14.150.1.3 free_client()

```
static void free_client (
    server_t * server ) [static]
Definition at line 12 of file free.c.
13 {
14     client_t *current_client = server->client;
15     client_t *next_client;
16
17     if (server->client->next != NULL)
18         current_client = server->client->next;
19     while (current_client != NULL) {
20         next_client = current_client->next;
21         free_node(current_client, server);
22         current_client = next_client;
23     }
24     server->client = NULL;
25 }
```

14.150.1.4 free_eggs()

```
static void free_eggs (
    server_t * server ) [static]
Definition at line 27 of file free.c.
28 {
29     egg_t *current_egg = server->eggs;
30     egg_t *next_egg;
31
32     while (current_egg != NULL) {
33         next_egg = current_egg->next;
34         free(current_egg->team_name);
35         current_egg->team_name = NULL;
36         free(current_egg);
37         current_egg = next_egg;
38     }
39     server->eggs = NULL;
40 }
```

14.150.1.5 free_map()

```
static void free_map (
    server_t * server,
    parsing_info_t * parsed_info ) [static]
Definition at line 53 of file free.c.
54 {
55     if (server->map && parsed_info && parsed_info->height > 0
56         && parsed_info->width > 0) {
57         for (int i = 0; i < parsed_info->height; i++) {
58             free_tiles(server, i, parsed_info);
59         }
60         free(server->map);
61         server->map = NULL;
62     }
63     free(server->serv_add);
64     server->serv_add = NULL;
65     free(server->total_resources);
```

```

66     server->total_resources = NULL;
67     free(server->current_resources);
68     server->current_resources = NULL;
69 }
```

14.150.1.6 free_parsed_info()

```
static void free_parsed_info (
    parsing_info_t * p_info ) [static]
```

Definition at line 71 of file free.c.

```

72 {
73     if (!p_info || !p_info->names)
74         return;
75     for (int i = 0; p_info->names[i] != NULL; i++) {
76         free(p_info->names[i]);
77         p_info->names[i] = NULL;
78     }
79     free(p_info->names);
80     p_info->names = NULL;
81 }
```

14.150.1.7 free_poll mana()

```
static void free_poll mana (
    server_t * server ) [static]
```

Definition at line 83 of file free.c.

```

84 {
85     if (server->poll_manager->fds)
86         free(server->poll_manager->fds);
87     free(server->poll_manager);
88 }
```

14.150.1.8 free_tiles()

```
static void free_tiles (
    server_t * server,
    int i,
    parsing_info_t * parsed_info ) [static]
```

Definition at line 42 of file free.c.

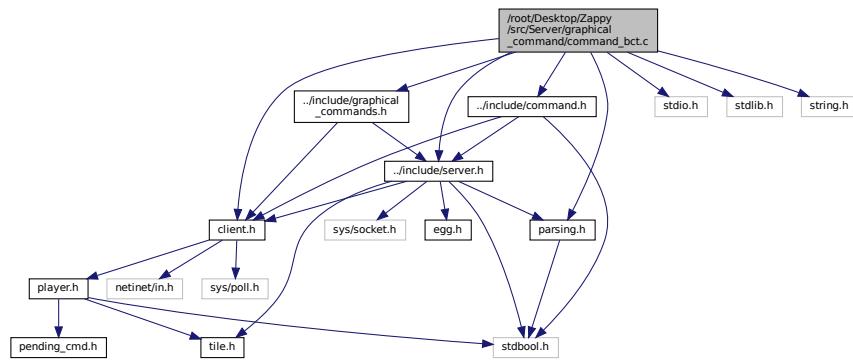
```

43 {
44     if (!server || !server->map || !server->map[i] || !parsed_info)
45         return;
46     for (int j = 0; j < parsed_info->width; j++) {
47         tile_free(&server->map[i][j]);
48     }
49     free(server->map[i]);
50     server->map[i] = NULL;
51 }
```

14.151 /root/Desktop/Zappy/src/Server/graphical_command/command_← bct.c File Reference

```
#include "../include/server.h"
#include "../include/client.h"
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include "../include/parsing.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Include dependency graph for command_bct.c:



Functions

- static int `calculate_size_bct_command` (int x, int y, `tile_t` *tile)
- static char * `get_buffer_bct_command` (int x, int y, `tile_t` *tile)
- static void `send_bct_command` (`server_t` *server, `client_t` *client, int x, int y)
- void `send_bct_to_all_graphical_clients` (`server_t` *server, int x, int y)
- void `send_mtc_to_all_graphical_clients` (`server_t` *server)
- void `send_tile_content_to_one_client` (`server_t` *server, `client_t` *client)
- void `command_bct` (`server_t` *server, `client_t` *client, char **buffer)
- void `command_mtc` (`server_t` *server, `client_t` *client, char **buffer)

14.151.1 Function Documentation

14.151.1.1 calculate_size_bct_command()

```

static int calculate_size_bct_command (
    int x,
    int y,
    tile_t * tile ) [static]
Definition at line 17 of file command_bct.c.
18 {
19     return sprintf(NULL, 0, "bct %d %d %d %d %d %d %d %d %d %d\n",
20                 x,
21                 y,
22                 tile->resources[FOOD],
23                 tile->resources[LINEMATE],
24                 tile->resources[DERAUMERE],
25                 tile->resources[SIBUR],
26                 tile->resources[MENDIANE],
27                 tile->resources[PHIRAS],
28                 tile->resources[THYSTAME]);
    
```

14.151.1.2 command_bct()

```

void command_bct (
    server_t * server,
    client_t * client,
    char ** buffer )
Definition at line 99 of file command_bct.c.
100 {
101     int x = 0;
102     int y = 0;
    
```

```

103
104     if (!server || !client || !buffer || !server->graphical_clients ||
105         arr_len(buffer) != 3 || sscanf(buffer[1], "%d", &x) != 1 ||
106         sscanf(buffer[2], "%d", &y) != 1 ||
107         x < 0 || y < 0 ||
108         y >= server->parsed_info->height ||
109         x >= server->parsed_info->width)
110     return write_command_output(client->client_fd, "sbp\n");
111 send_bct_command(server, client, x, y);
112 }

```

14.151.1.3 command_mtc()

```

void command_mtc (
    server_t * server,
    client_t * client,
    char ** buffer )

```

Definition at line 114 of file command_bct.c.

```

115 {
116     if (!server || !client || !server->graphical_clients
117         || arr_len(buffer) != 1)
118     return write_command_output(client->client_fd, "sbp\n");
119     send_tile_content_to_one_client(server, client);
120 }

```

14.151.1.4 get_buffer_bct_command()

```

static char* get_buffer_bct_command (
    int x,
    int y,
    tile_t * tile ) [static]

```

Definition at line 30 of file command_bct.c.

```

31 {
32     int size = calculate_size_bct_command(x, y, tile);
33     char *buffer = malloc(size + 1);
34
35     if (!buffer)
36         return NULL;
37     snprintf(buffer, size + 1, "bct %d %d %d %d %d %d %d %d %d %d\n",
38             x, y,
39             tile->resources[FOOD],
40             tile->resources[LINEMATE],
41             tile->resources[DERAUMERE],
42             tile->resources[SIBUR],
43             tile->resources[MENDIANE],
44             tile->resources[PHIRAS],
45             tile->resources[THYSTAME]);
46     return buffer;
47 }

```

14.151.1.5 send_bct_command()

```

static void send_bct_command (
    server_t * server,
    client_t * client,
    int x,
    int y ) [static]

```

Definition at line 49 of file command_bct.c.

```

50 {
51     tile_t *tile = NULL;
52     char *buffer = NULL;
53
54     if (!server || !client ||
55         x < 0 || y < 0 ||
56         y >= server->parsed_info->height || x >= server->parsed_info->width)
57     return;
58     tile = &server->map[y][x];
59     buffer = get_buffer_bct_command(x, y, tile);
60     write_command_output(client->client_fd, buffer);
61     free(buffer);
62 }

```

14.151.1.6 send_bct_to_all_graphical_clients()

```
void send_bct_to_all_graphical_clients (
    server_t * server,
    int x,
    int y )
```

Definition at line 64 of file command_bct.c.

```
65 {
66     graphical_client_t *current = server->graphical_clients;
67
68     if (!server || !server->graphical_clients)
69         return;
70     while (current) {
71         send_bct_command(server, current->client, x, y);
72         current = current->next;
73     }
74 }
```

14.151.1.7 send_mtc_to_all_graphical_clients()

```
void send_mtc_to_all_graphical_clients (
    server_t * server )
```

Definition at line 76 of file command_bct.c.

```
77 {
78     graphical_client_t *current = server->graphical_clients;
79
80     if (!server || !server->graphical_clients)
81         return;
82     while (current) {
83         send_tile_content_to_one_client(server, current->client);
84         current = current->next;
85     }
86 }
```

14.151.1.8 send_tile_content_to_one_client()

```
void send_tile_content_to_one_client (
    server_t * server,
    client_t * client )
```

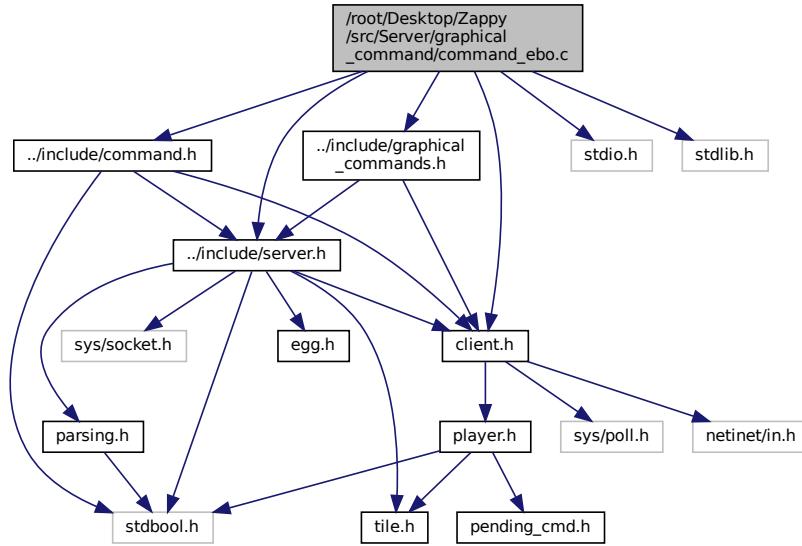
Definition at line 88 of file command_bct.c.

```
89 {
90     if (!server || !client || !server->graphical_clients)
91         return;
92     for (int y = 0; y < server->parsed_info->height; y++) {
93         for (int x = 0; x < server->parsed_info->width; x++) {
94             send_bct_command(server, client, x, y);
95         }
96     }
97 }
```

14.152 /root/Desktop/Zappy/src/Server/graphical_command/command_ebo.c File Reference

```
#include "../include/server.h"
#include "../include/client.h"
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include <stdio.h>
#include <stdlib.h>
```

Include dependency graph for command_ebo.c:



Functions

- void [send_ebo_command](#) (server_t *server, int egg_id)

14.152.1 Function Documentation

14.152.1.1 send_ebo_command()

```
void send_ebo_command (
    server_t * server,
    int egg_id )
```

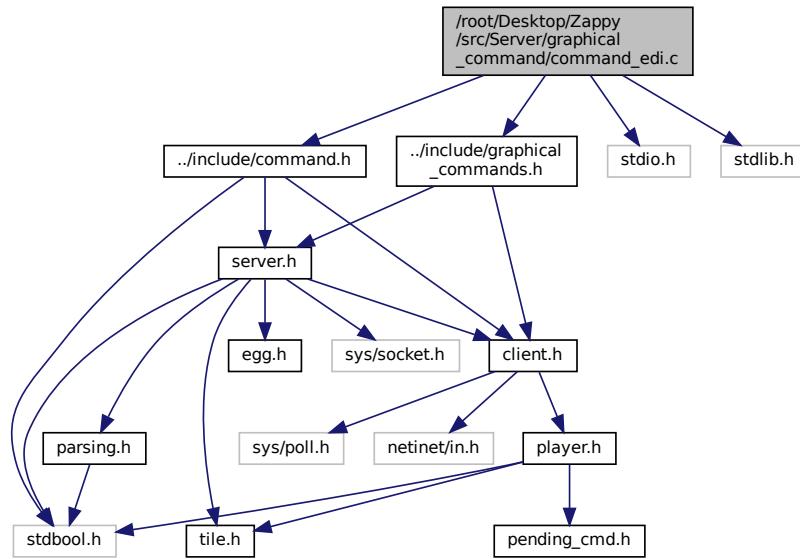
Definition at line 15 of file command_ebo.c.

```
16 {
17     graphical_client_t *current = NULL;
18     char *buffer;
19     int size;
20
21     if (!server || !server->graphical_clients)
22         return;
23     size = snprintf(NULL, 0, "ebo #%d\n", egg_id);
24     buffer = malloc(size + 1);
25     if (!buffer)
26         server_err("Buffer alloc in ebo command failed\n");
27     sprintf(buffer, "ebo #%d\n", egg_id);
28     current = server->graphical_clients;
29     while (current) {
30         write_command_output(current->client->client_fd, buffer);
31         current = current->next;
32     }
33     free(buffer);
34 }
```

14.153 /root/Desktop/Zappy/src/Server/graphical_command/command_edi.c File Reference

```
#include "../include/command.h"
#include "../include/graphical_commands.h"
```

```
#include <stdio.h>
#include <stdlib.h>
Include dependency graph for command_edi.c:
```



Functions

- void [send_edi_command \(server_t *server, int egg_id\)](#)

14.153.1 Function Documentation

14.153.1.1 send_edi_command()

```
void send_edi_command (
    server_t * server,
    int egg_id )
```

Definition at line 13 of file command_edi.c.

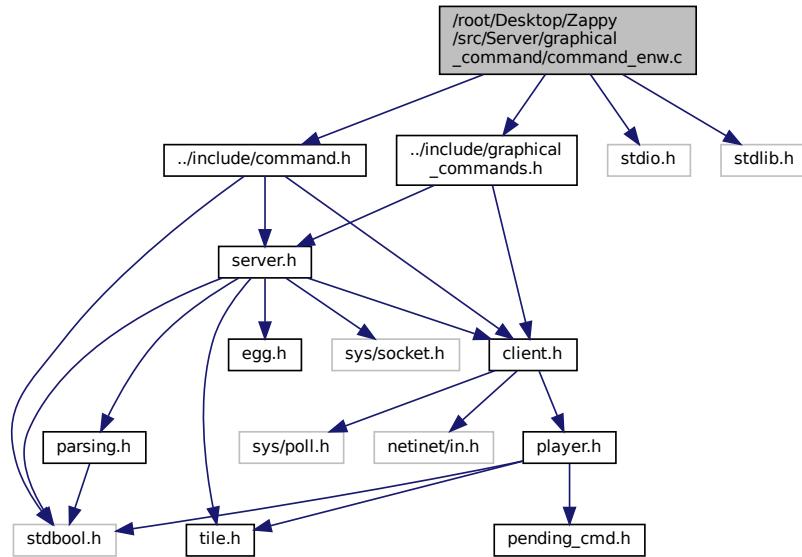
```

14 {
15     char *buffer = NULL;
16     int size = 0;
17     graphical_client_t *graphical_client = NULL;
18
19     if (!server || !server->graphical_clients)
20         return;
21     size = snprintf(NULL, 0, "edi #%d\n", egg_id);
22     buffer = malloc(size + 1);
23     if (!buffer)
24         return;
25     snprintf(buffer, size + 1, "edi #%d\n", egg_id);
26     graphical_client = server->graphical_clients;
27     while (graphical_client) {
28         write_command_output(graphical_client->client->client_fd, buffer);
29         graphical_client = graphical_client->next;
30     }
31     free(buffer);
32 }
```

14.154 /root/Desktop/Zappy/src/Server/graphical_command/command_enw.c File Reference

```
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include <stdio.h>
#include <stdlib.h>
```

Include dependency graph for command_enw.c:



Functions

- static char * [get_buffer_for_enw](#) (int egg_id, int client_id, int pos_x, int pos_y)
- void [send_enw_command](#) (server_t *server, client_t *client, int egg_id)
- static void [send_enw_command_to_client](#) (client_t *client, egg_t *egg)
- void [send_enw_command_start](#) (server_t *server)

14.154.1 Function Documentation

14.154.1.1 get_buffer_for_enw()

```
static char* get_buffer_for_enw (
    int egg_id,
    int client_id,
    int pos_x,
    int pos_y ) [static]
```

Definition at line 13 of file command_enw.c.

```
15 {
16     char *buffer = NULL;
17     int size = sprintf(NULL, 0, "enw #%d %#d %d %d\n",
18                         egg_id, client_id, pos_x, pos_y);
19
20     buffer = malloc(size + 1);
21     if (buffer) {
22         sprintf(buffer, size + 1, "enw #%d %#d %d %d\n",
23                 egg_id, client_id, pos_x, pos_y);
24     }
}
```

```
25     return buffer;
26 }
```

14.154.1.2 send_enw_command()

```
void send_enw_command (
    server_t * server,
    client_t * client,
    int egg_id )
```

Definition at line 28 of file command_enw.c.

```
29 {
30     char *buffer = NULL;
31     graphical_client_t *graphical_client = NULL;
32
33     if (!server || !client || !client->player || !server->graphical_clients)
34         return;
35     buffer = get_buffer_for_enw(egg_id, client->client_id,
36         client->player->pos_x, client->player->pos_y);
37     if (!buffer)
38         return;
39     graphical_client = server->graphical_clients;
40     while (graphical_client) {
41         write_command_output(graphical_client->client->client_fd, buffer);
42         graphical_client = graphical_client->next;
43     }
44     free(buffer);
45 }
```

14.154.1.3 send_enw_command_start()

```
void send_enw_command_start (
    server_t * server )
```

Definition at line 60 of file command_enw.c.

```
61 {
62     graphical_client_t *graphical_client;
63     egg_t *egg;
64
65     if (!server || !server->graphical_clients)
66         return;
67     egg = server->eggs;
68     while (egg) {
69         graphical_client = server->graphical_clients;
70         while (graphical_client) {
71             send_enw_command_to_client(graphical_client->client, egg);
72             graphical_client = graphical_client->next;
73         }
74         egg = egg->next;
75     }
76 }
```

14.154.1.4 send_enw_command_to_client()

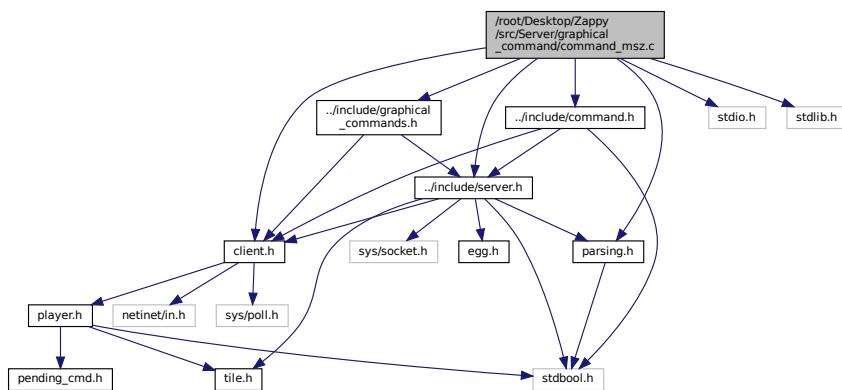
```
static void send_enw_command_to_client (
    client_t * client,
    egg_t * egg ) [static]
```

Definition at line 47 of file command_enw.c.

```
49 {
50     char *buffer = NULL;
51
52     buffer = get_buffer_for_enw(egg->egg_id, -1,
53         egg->pos_x, egg->pos_y);
54     if (buffer) {
55         write_command_output(client->client_fd, buffer);
56         free(buffer);
57     }
58 }
```

14.155 /root/Desktop/Zappy/src/Server/graphical_command/command_msz.c File Reference

```
#include "../include/server.h"
#include "../include/client.h"
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include "../include/parsing.h"
#include <stdio.h>
#include <stdlib.h>
Include dependency graph for command_msz.c:
```



Functions

- void `send_msz_command (server_t *server, client_t *client)`
- void `command_msz (server_t *server, client_t *client, char **buffer)`

14.155.1 Function Documentation

14.155.1.1 command_msz()

```
void command_msz (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 35 of file command_msz.c.

```
36 {
37     if (!server || !client || !server->graphical_clients ||
38         arr_len(buffer) != 1)
39     return write_command_output(client->client_fd, "sbp\n");
40     send_msz_command(server, client);
41 }
```

14.155.1.2 send_msz_command()

```
void send_msz_command (
    server_t * server,
    client_t * client )
```

Definition at line 16 of file command_msz.c.

```
17 {
18     char *buffer = NULL;
```

```

19     int size = 0;
20
21     if (!server || !client)
22         return;
23     buffer = NULL;
24     size = sprintf(NULL, 0, "msz %d %d\n",
25                     server->parsed_info->width, server->parsed_info->height);
26     buffer = malloc(size + 1);
27     if (!buffer)
28         return;
29     sprintf(buffer, size + 1, "msz %d %d\n",
30             server->parsed_info->width, server->parsed_info->height);
31     write_command_output(client->client_fd, buffer);
32     free(buffer);
33 }

```

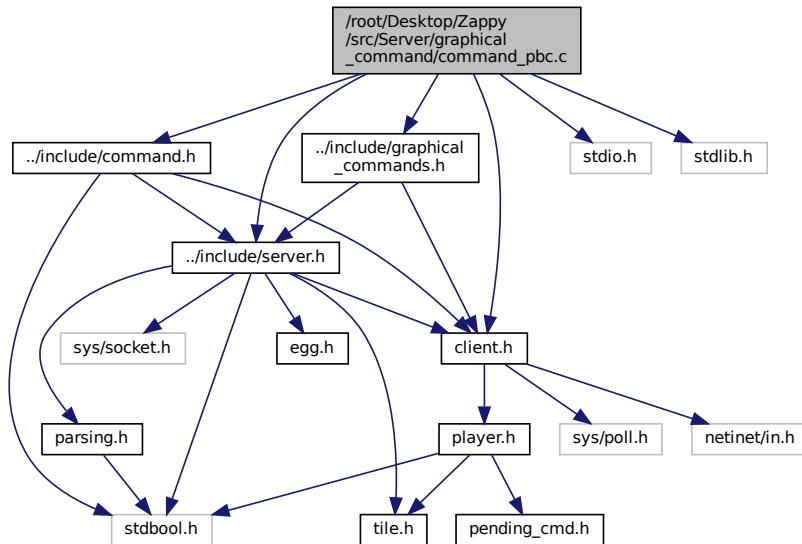
14.156 /root/Desktop/Zappy/src/Server/graphical_command/command_pbc.c File Reference

```

#include "../include/server.h"
#include "../include/client.h"
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include <stdio.h>
#include <stdlib.h>

```

Include dependency graph for command_pbc.c:



Functions

- void [command_pbc \(server_t *server, client_t *client, char *buffer\)](#)

14.156.1 Function Documentation

14.156.1.1 command_pbc()

```

void command_pbc (
    server_t * server,

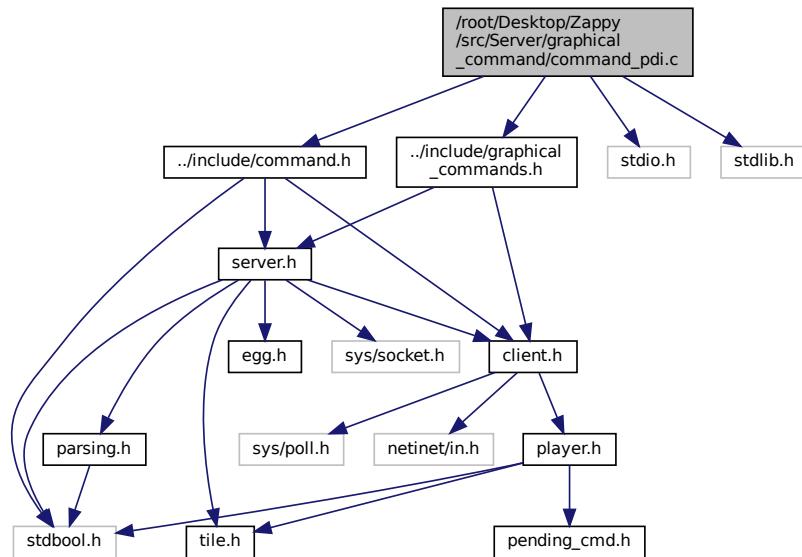
```

```

    client_t * client,
    char * buffer )
Definition at line 15 of file command_pbc.c.
16 {
17     graphical_client_t *graphical_client = NULL;
18     char *msg = NULL;
19     int size = 0;
20
21     if (!server || !client || !buffer ||
22         !server->graphical_clients || !client->player)
23         return;
24     size = sprintf(NULL, 0, "pbc %d %s", client->client_id, buffer);
25     msg = malloc(size + 1);
26     if (!msg)
27         return;
28     sprintf(msg, size + 1, "pbc %d %s", client->client_id, buffer);
29     graphical_client = server->graphical_clients;
30     while (graphical_client) {
31         write_command_output(graphical_client->client->client_fd, msg);
32         graphical_client = graphical_client->next;
33     }
34     free(msg);
35 }
```

14.157 /root/Desktop/Zappy/src/Server/graphical_command/command_pdi.c File Reference

```
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include <stdio.h>
#include <stdlib.h>
Include dependency graph for command_pdi.c:
```



Functions

- void `command_pdi (server_t *server, client_t *client)`

14.157.1 Function Documentation

14.157.1.1 command_pdi()

```
void command_pdi (
    server_t * server,
    client_t * client )
```

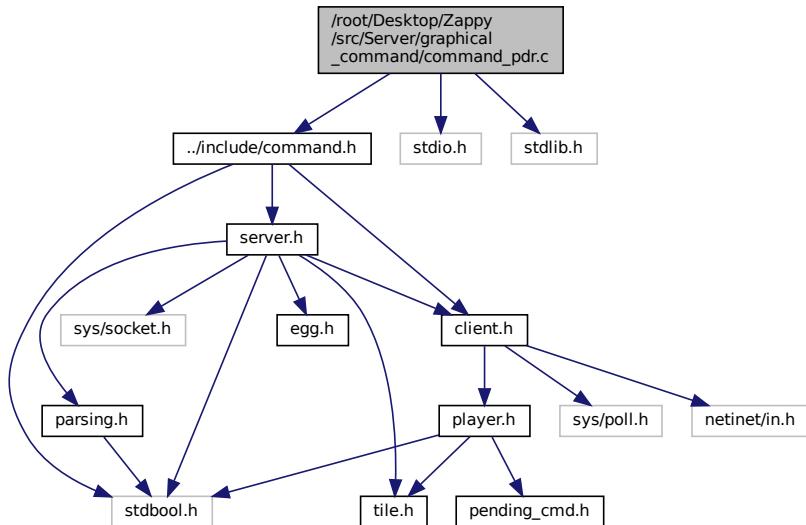
Definition at line 13 of file command_pdi.c.

```
14 {
15     graphical_client_t *graphical_client = NULL;
16     char *buffer = NULL;
17     int size = 0;
18
19     if (!client || !client->player || !server->graphical_clients)
20         return;
21     size = sprintf(NULL, 0, "pdi %d\n", client->client_id);
22     buffer = malloc(size + 1);
23     if (buffer == NULL)
24         return;
25     sprintf(buffer, "pdi %d\n", client->client_id);
26     graphical_client = server->graphical_clients;
27     while (graphical_client != NULL) {
28         write_command_output(graphical_client->client->client_fd,
29             buffer);
30         graphical_client = graphical_client->next;
31     }
32     free(buffer);
33 }
```

14.158 /root/Desktop/Zappy/src/Server/graphical_command/command_pdr.c File Reference

```
#include "../include/command.h"
#include <stdio.h>
#include <stdlib.h>
```

Include dependency graph for command_pdr.c:



Functions

- void `command_pdr (server_t *server, client_t *client, resource_type_t resource_type)`

14.158.1 Function Documentation

14.158.1.1 command_pdr()

```
void command_pdr (
    server_t * server,
    client_t * client,
    resource_type_t resource_type )
```

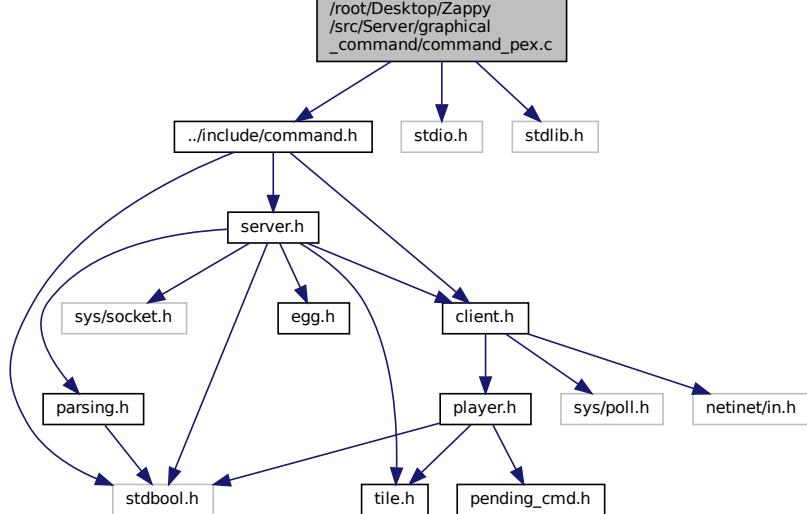
Definition at line 12 of file command_pdr.c.

```
14 {
15     graphical_client_t *graphical_client = NULL;
16     char *buffer = NULL;
17     int size = 0;
18
19     if (!client || !client->player || !server->graphical_clients)
20         return;
21     size = snprintf(NULL, 0, "pdr %d %d\n", client->client_id, resource_type);
22     buffer = malloc(size + 1);
23     if (buffer == NULL)
24         return write_command_output(client->client_fd, "ko\n");
25     sprintf(buffer, "pdr %d %d\n", client->client_id, resource_type);
26     graphical_client = server->graphical_clients;
27     while (graphical_client != NULL) {
28         write_command_output(graphical_client->client->client_fd, buffer);
29         graphical_client = graphical_client->next;
30     }
31     free(buffer);
32 }
```

14.159 /root/Desktop/Zappy/src/Server/graphical_command/command_pex.c File Reference

```
#include "../include/command.h"
#include <stdio.h>
#include <stdlib.h>
```

Include dependency graph for command_pex.c:



Functions

- void command_pex (server_t *server, client_t *client)

14.159.1 Function Documentation

14.159.1.1 command_pex()

```
void command_pex (
    server_t * server,
    client_t * client )
```

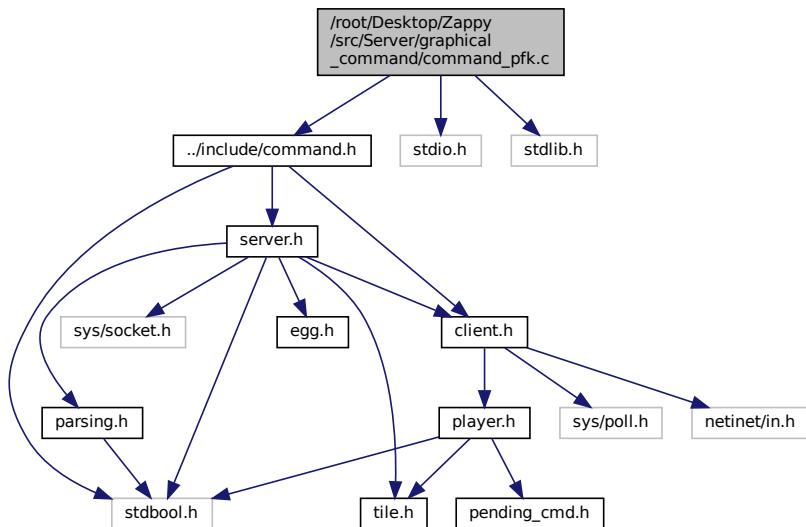
Definition at line 12 of file command_pex.c.

```
13 {
14     graphical_client_t *graphical_client = NULL;
15     char *buffer = NULL;
16     int size = 0;
17
18     if (!client || !client->player || !server->graphical_clients)
19         return;
20     size = sprintf(NULL, 0, "pex %d\n", client->client_id);
21     buffer = malloc(size + 1);
22     if (buffer == NULL)
23         return write_command_output(client->client_fd, "ko\n");
24     sprintf(buffer, "pex %d\n", client->client_id);
25     graphical_client = server->graphical_clients;
26     while (graphical_client != NULL) {
27         write_command_output(graphical_client->client->client_fd,
28                             buffer);
29         graphical_client = graphical_client->next;
30     }
31     free(buffer);
32 }
```

14.160 /root/Desktop/Zappy/src/Server/graphical_command/command_pfk.c File Reference

```
#include "../include/command.h"
#include <stdio.h>
#include <stdlib.h>
```

Include dependency graph for command_pfk.c:



Functions

- void command_pfk (server_t *server, client_t *client)

14.160.1 Function Documentation

14.160.1.1 command_pfk()

```
void command_pfk (
    server_t * server,
    client_t * client )
```

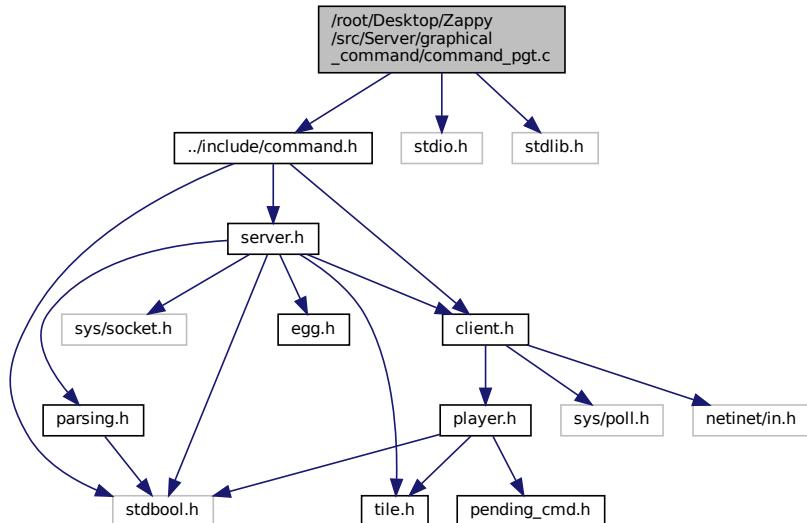
Definition at line 12 of file command_pfk.c.

```
13 {
14     graphical_client_t *graphical_client = NULL;
15     char *buffer = NULL;
16     int size = 0;
17
18     if (!client || !client->player || !server->graphical_clients)
19         return;
20     size = sprintf(NULL, 0, "pfk #%d\n", client->client_id);
21     buffer = malloc(size + 1);
22     if (buffer == NULL)
23         return write_command_output(client->client_fd, "ko\n");
24     sprintf(buffer, "pfk #%d\n", client->client_id);
25     graphical_client = server->graphical_clients;
26     while (graphical_client != NULL) {
27         write_command_output(graphical_client->client->client_fd, buffer);
28         graphical_client = graphical_client->next;
29     }
30     free(buffer);
31 }
```

14.161 /root/Desktop/Zappy/src/Server/graphical_command/command_pgt.c File Reference

```
#include "../include/command.h"
#include <stdio.h>
#include <stdlib.h>
```

Include dependency graph for command_pgt.c:



Functions

- void `command_pgt (server_t *server, client_t *client, resource_type_t resource_type)`

14.161.1 Function Documentation

14.161.1.1 command_pgt()

```
void command_pgt (
    server_t * server,
    client_t * client,
    resource_type_t resource_type )
```

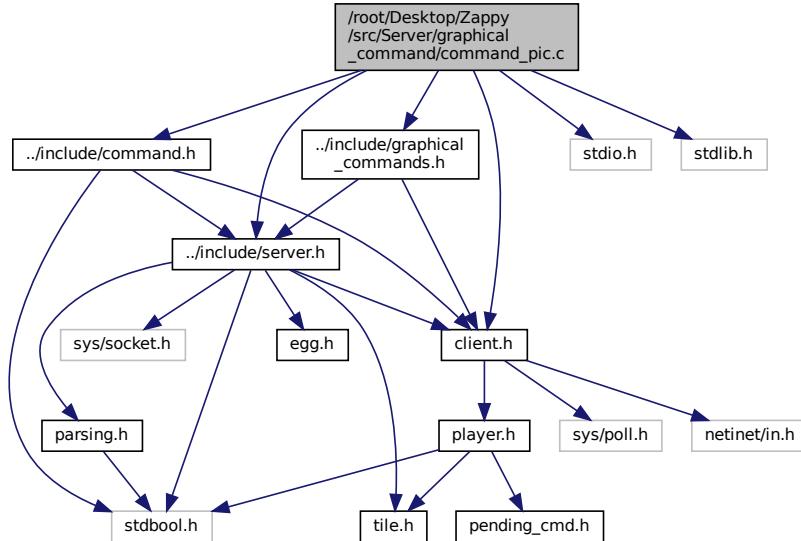
Definition at line 12 of file command_pgt.c.

```
14 {
15     graphical_client_t *graphical_client = NULL;
16     char *buffer = NULL;
17     int size = 0;
18
19     if (!client || !client->player || !server->graphical_clients)
20         return;
21     size = sprintf(NULL, 0, "pgt #%d %d\n", client->client_id, resource_type);
22     buffer = malloc(size + 1);
23     if (buffer == NULL)
24         return;
25     sprintf(buffer, "pgt #%d %d\n", client->client_id, resource_type);
26     graphical_client = server->graphical_clients;
27     while (graphical_client != NULL) {
28         write_command_output(graphical_client->client->client_fd, buffer);
29         graphical_client = graphical_client->next;
30     }
31     free(buffer);
32 }
```

14.162 /root/Desktop/Zappy/src/Server/graphical_command/command_pic.c File Reference

```
#include "../include/server.h"
#include "../include/client.h"
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include <stdio.h>
#include <stdlib.h>
```

Include dependency graph for command_pic.c:



Functions

- static char * get_pic_buffer (int x, int y, int level, tile_t *tile)

- void `command_pic (server_t *server, int x, int y, int level)`

14.162.1 Function Documentation

14.162.1.1 command_pic()

```
void command_pic (
    server_t * server,
    int x,
    int y,
    int level )
```

Definition at line 36 of file command_pic.c.

```
37 {
38     graphical_client_t *current = NULL;
39     char *buffer = NULL;
40     tile_t *tile = NULL;
41
42     if (!server || !server->graphical_clients)
43         return;
44     if (y < 0 || x < 0 || y >= server->parsed_info->height
45         || x >= server->parsed_info->width)
46         return;
47     tile = &server->map[y][x];
48     buffer = get_pic_buffer(x, y, level, tile);
49     if (!buffer)
50         return;
51     current = server->graphical_clients;
52     while (current) {
53         write_command_output(current->client->client_fd, buffer);
54         current = current->next;
55     }
56     free(buffer);
57 }
```

14.162.1.2 get_pic_buffer()

```
static char* get_pic_buffer (
    int x,
    int y,
    int level,
    tile_t * tile ) [static]
```

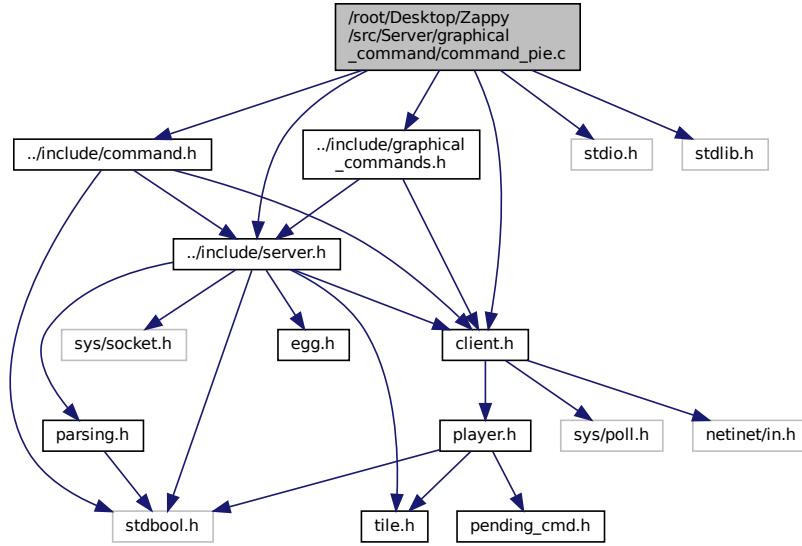
Definition at line 15 of file command_pic.c.

```
16 {
17     int size = 0;
18     char *buffer = NULL;
19     int offset = 0;
20
21     size = sprintf(NULL, 0, "pic %d %d %d", x, y, level);
22     for (int i = 0; i < tile->player_count; i++)
23         size += sprintf(NULL, 0, " #%d", tile->player_ids[i]);
24     size += 2;
25     buffer = malloc(size);
26     if (!buffer)
27         return NULL;
28     offset = sprintf(buffer, size, "pic %d %d %d", x, y, level);
29     for (int i = 0; i < tile->player_count; i++)
30         offset += sprintf(buffer + offset, size - offset, " #%d",
31                           tile->player_ids[i]);
32     sprintf(buffer + offset, size - offset, "\n");
33     return buffer;
34 }
```

14.163 /root/Desktop/Zappy/src/Server/graphical_command/command_pie.c File Reference

```
#include "../include/server.h"
#include "../include/client.h"
#include "../include/command.h"
```

```
#include "../include/graphical_commands.h"
#include <stdio.h>
#include <stdlib.h>
Include dependency graph for command_pie.c:
```



Functions

- static char * [get_pie_buffer](#) (int x, int y, int result)
- void [command_pie](#) ([server_t](#) *server, int x, int y, int result)

14.163.1 Function Documentation

14.163.1.1 [command_pie\(\)](#)

```
void command_pie (
    server\_t * server,
    int x,
    int y,
    int result )
```

Definition at line 28 of file command_pie.c.

```
29 {
30     client\_t *current = NULL;
31     char *buffer = NULL;
32
33     if (!server || !server->graphical_clients)
34         return;
35     buffer = get\_pie\_buffer(x, y, result);
36     if (!buffer)
37         return;
38     current = server->client;
39     while (current) {
40         write\_command\_output(current->client_fd, buffer);
41         current = current->next;
42     }
43     free(buffer);
44 }
```

14.163.1.2 get_pie_buffer()

```
static char* get_pie_buffer (
    int x,
    int y,
    int result ) [static]
```

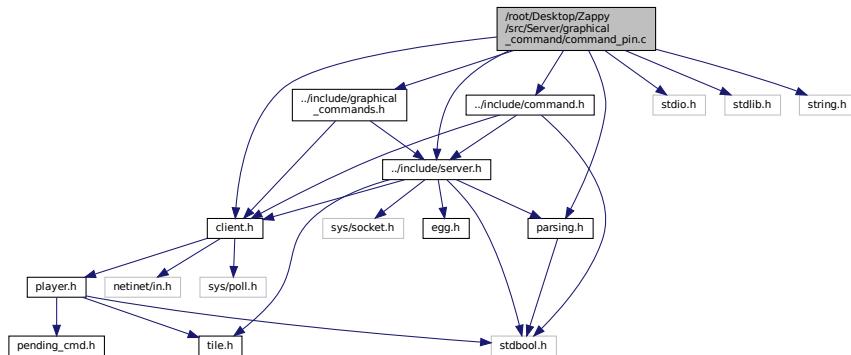
Definition at line 15 of file command_pie.c.

```
16 {
17     char *buffer = NULL;
18     int size = 0;
19
20     size = snprintf(NULL, 0, "pie %d %d %d\n", x, y, result);
21     buffer = malloc(size + 1);
22     if (!buffer)
23         return NULL;
24     snprintf(buffer, size + 1, "pie %d %d %d\n", x, y, result);
25     return buffer;
26 }
```

14.164 /root/Desktop/Zappy/src/Server/graphical_command/command_pin.c File Reference

```
#include "../include/server.h"
#include "../include/client.h"
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include "../include/parsing.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Include dependency graph for command_pin.c:



Functions

- static int calculate_size_pin_command (client_t *client)
- static char * get_buffer_pin_command (client_t *client)
- void send_pin_command (server_t *server, client_t *client, client_t *recipient)
- void send_pin_to_all (server_t *server, client_t *client)
- void command_pin (server_t *server, client_t *client, char **buffer)

14.164.1 Function Documentation

14.164.1.1 calculate_size_pin_command()

```
static int calculate_size_pin_command (
    client_t * client ) [static]
Definition at line 17 of file command_pin.c.
18 {
19     if (!client || !client->player)
20         return 0;
21     return sprintf(NULL, 0, "pin %d %d %d %d %d %d %d %d %d %d\n",
22                 client->client_id,
23                 client->player->pos_x,
24                 client->player->pos_y,
25                 client->player->inventory[FOOD],
26                 client->player->inventory[LINEMATE],
27                 client->player->inventory[DERAUMERE],
28                 client->player->inventory[SIBUR],
29                 client->player->inventory[MENDIANE],
30                 client->player->inventory[PHIRAS],
31                 client->player->inventory[THYSTAME]);
32 }
```

14.164.1.2 command_pin()

```
void command_pin (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 78 of file command_pin.c.

```
79 {
80     client_t *recipient = NULL;
81     int id = -1;
82
83     if (!server || !client || !server->graphical_clients ||
84         arr_len(buffer) != 2 ||
85         sscanf(buffer[1], "#%d\n", &id) != 1 || id < 0)
86         return write_command_output(client->client_fd, "sbp\n");
87     recipient = find_client_by_id(server, id);
88     if (!recipient || recipient->type != AI) {
89         write_command_output(client->client_fd, "sbp\n");
90         return;
91     }
92     send_pin_command(server, recipient, client);
93 }
```

14.164.1.3 get_buffer_pin_command()

```
static char* get_buffer_pin_command (
    client_t * client ) [static]
```

Definition at line 34 of file command_pin.c.

```
35 {
36     int size = calculate_size_pin_command(client);
37     char *buffer = malloc(size + 1);
38
39     if (!buffer)
40         return NULL;
41     sprintf(buffer, size + 1, "pin %d %d %d %d %d %d %d %d %d %d\n",
42             client->client_id,
43             client->player->pos_x,
44             client->player->pos_y,
45             client->player->inventory[FOOD],
46             client->player->inventory[LINEMATE],
47             client->player->inventory[DERAUMERE],
48             client->player->inventory[SIBUR],
49             client->player->inventory[MENDIANE],
50             client->player->inventory[PHIRAS],
51             client->player->inventory[THYSTAME]);
52     return buffer;
53 }
```

14.164.1.4 send_pin_command()

```
void send_pin_command (
    server_t * server,
```

```

    client_t * client,
    client_t * recipient )
Definition at line 55 of file command_pin.c.
56 {
57     char *buffer = get_buffer_pin_command(client);
58
59     (void)server;
60     if (!buffer || !server->graphical_clients)
61         return;
62     write_command_output(recipient->client_fd, buffer);
63     free(buffer);
64 }

```

14.164.1.5 send_pin_to_all()

```

void send_pin_to_all (
    server_t * server,
    client_t * client )

```

Definition at line 66 of file command_pin.c.

```

67 {
68     graphical_client_t *current = server->graphical_clients;
69
70     if (!server || !client || !server->graphical_clients)
71         return;
72     while (current) {
73         send_pin_command(server, client, current->client);
74         current = current->next;
75     }
76 }

```

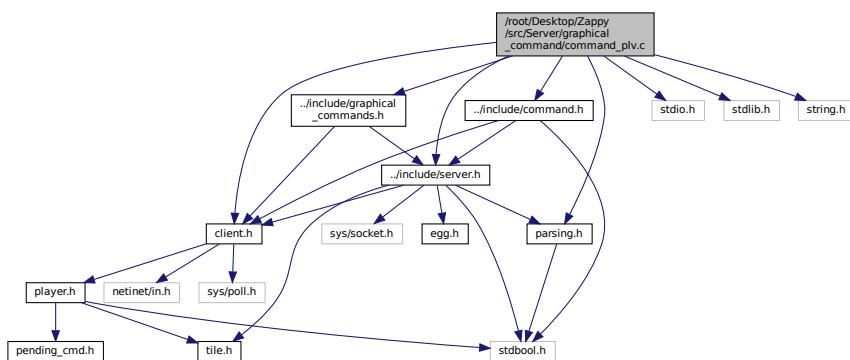
14.165 /root/Desktop/Zappy/src/Server/graphical_command/command_plv.c File Reference

```

#include "../include/server.h"
#include "../include/client.h"
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include "../include/parsing.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

```

Include dependency graph for command_plv.c:



Functions

- void `send_plv_command (server_t *server, client_t *client, client_t *recipient)`
- void `send_plv_to_all (server_t *server, client_t *client)`

- void `command_plv (server_t *server, client_t *client, char **buffer)`

14.165.1 Function Documentation

14.165.1.1 command_plv()

```
void command_plv (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 51 of file command_plv.c.

```
52 {
53     client_t *recipient = NULL;
54     int id = -1;
55
56     if (!server || !client || !buffer ||
57         !server->graphical_clients || arl_len(buffer) != 2 ||
58         sscanf(buffer[1], "#%d\n", &id) != 1)
59         return write_command_output(client->client_fd, "sbp\n");
60     recipient = find_client_by_id(server, id);
61     if (!recipient || recipient->type != AI) {
62         write_command_output(client->client_fd, "sbp\n");
63         return;
64     }
65     send_plv_command(server, recipient, client);
66 }
```

14.165.1.2 send_plv_command()

```
void send_plv_command (
    server_t * server,
    client_t * client,
    client_t * recipient )
```

Definition at line 17 of file command_plv.c.

```
18 {
19     int size = 0;
20     char *buffer = NULL;
21
22     if (!server || !client || !client->player)
23         return;
24     size = sprintf(NULL, 0, "plv #%d %d\n",
25                   client->client_id,
26                   client->player->level);
27     buffer = malloc(size + 1);
28     if (!buffer)
29         return;
30     snprintf(buffer, size + 1, "plv #%d %d\n",
31               client->client_id,
32               client->player->level);
33     write_command_output(recipient->client_fd, buffer);
34     free(buffer);
35 }
```

14.165.1.3 send_plv_to_all()

```
void send_plv_to_all (
    server_t * server,
    client_t * client )
```

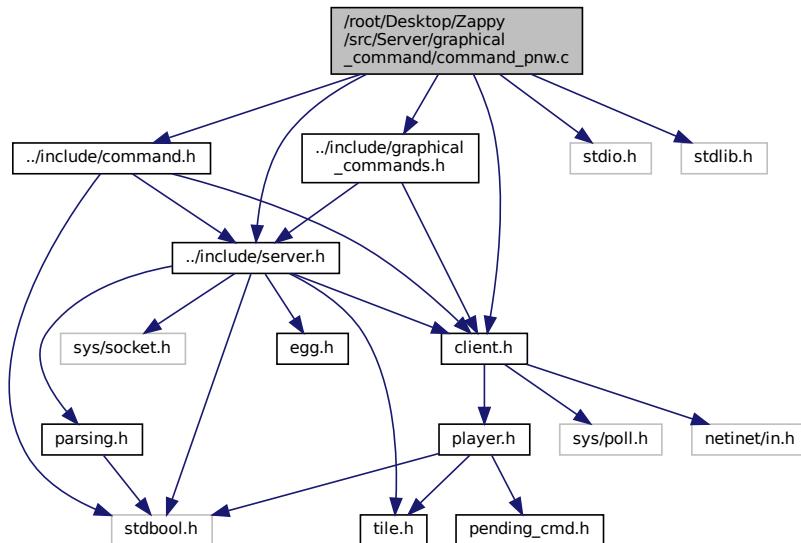
Definition at line 37 of file command_plv.c.

```
38 {
39     client_t *current = server->client;
40
41     if (!server || !client || !client->player)
42         return;
43     while (current) {
44         if (current->type == GRAPHICAL && current != client) {
45             send_plv_command(server, client, current);
46         }
47         current = current->next;
```

```
48      }
49 }
```

14.166 /root/Desktop/Zappy/src/Server/graphical_command/command_pnw.c File Reference

```
#include "../include/server.h"
#include "../include/client.h"
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include <stdio.h>
#include <stdlib.h>
Include dependency graph for command_pnw.c:
```



Functions

- static int `calcul_size_pnw_command (client_t *client)`
- void `send_pnw_command (server_t *server, client_t *client, client_t *recipient)`
- void `send_pnw_command_to_all (server_t *server, client_t *client)`
- void `send_all_player_info_to_one_client (server_t *server, client_t *client)`

14.166.1 Function Documentation

14.166.1.1 calcul_size_pnw_command()

```
static int calcul_size_pnw_command (
    client_t * client ) [static]
Definition at line 15 of file command_pnw.c.
16 {
17     if (!client || !client->player)
18         return 0;
19     return snprintf(NULL, 0, "pnw %d %d %d %d %d %s\n",
20                     client->client_id,
21                     client->player->pos_x,
22                     client->player->pos_y,
```

```

23         client->player->rotation,
24         client->player->level,
25         client->player->team_name);
26 }

```

14.166.1.2 send_all_player_info_to_one_client()

```

void send_all_player_info_to_one_client (
    server_t * server,
    client_t * client )

```

Definition at line 64 of file command_pnw.c.

```

65 {
66     client_t *current = NULL;
67
68     if (!server || !client || !server->graphical_clients)
69         return;
70     current = server->client;
71     while (current) {
72         if (current->player && current->type == AI &&
73             current->is_fully_connected) {
74             send_pnw_command(server, current, client);
75             send_pin_command(server, current, client);
76             send_plv_command(server, current, client);
77         }
78         current = current->next;
79     }
80 }

```

14.166.1.3 send_pnw_command()

```

void send_pnw_command (
    server_t * server,
    client_t * client,
    client_t * recipient )

```

Definition at line 28 of file command_pnw.c.

```

29 {
30     int size = 0;
31     char *buffer = NULL;
32
33     if (!server || !client || !client->player)
34         return;
35     size = calcul_size_pnw_command(client);
36     buffer = malloc(size + 1);
37     if (!buffer)
38         return;
39     snprintf(buffer, size + 1, "pnw %d %d %d %d %d %s\n",
40             client->client_id,
41             client->player->pos_x,
42             client->player->pos_y,
43             client->player->rotation,
44             client->player->level,
45             client->player->team_name);
46     write_command_output(recipient->client_fd, buffer);
47     free(buffer);
48 }

```

14.166.1.4 send_pnw_command_to_all()

```

void send_pnw_command_to_all (
    server_t * server,
    client_t * client )

```

Definition at line 50 of file command_pnw.c.

```

51 {
52     client_t *current = NULL;
53
54     if (!server || !client || !client->player)
55         return;
56     current = server->client;
57     while (current) {
58         if (current->type == GRAPHICAL && current->is_fully_connected)
59             send_pnw_command(server, client, current);
60         current = current->next;

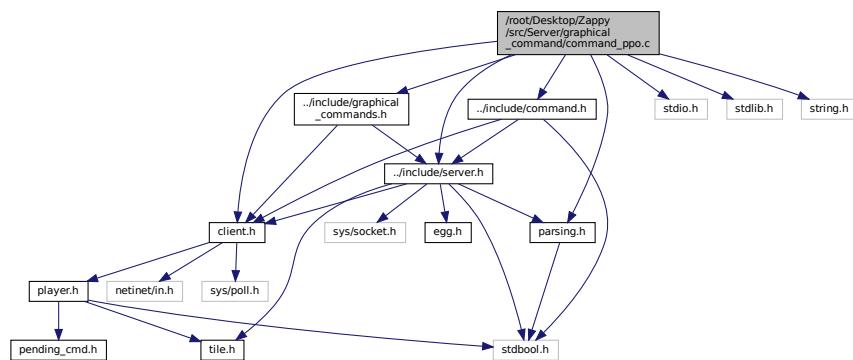
```

```
61     }
62 }
```

14.167 /root/Desktop/Zappy/src/Server/graphical_command/command_ppo.c File Reference

```
#include "../include/server.h"
#include "../include/client.h"
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include "../include/parsing.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Include dependency graph for command_ppo.c:



Functions

- `client_t * find_client_by_id (server_t *server, int id)`
- `static char * get_ppo_buffer (client_t *tmp)`
- `bool send_ppo_command (server_t *server, int id)`
- `void command_ppo (server_t *server, client_t *client, char **buffer)`

14.167.1 Function Documentation

14.167.1.1 command_ppo()

```
void command_ppo (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 68 of file command_ppo.c.

```
69 {
70     int id = 0;
71
72     if (!server || !client || !buffer || !server->graphical_clients ||
73         arr_len(buffer) != 2 || sscanf(buffer[1], "#%d\n", &id) != 1 ||
74         id < 0 || !find_client_by_id(server, id) ||
75         !send_ppo_command(server, id)) {
76         return write_command_output(client->client_fd, "sbp\n");
77     }
78 }
```

14.167.1.2 find_client_by_id()

```
client_t* find_client_by_id (
    server_t * server,
    int id )
```

Definition at line 17 of file command_ppo.c.

```
18 {
19     client_t *current = server->client;
20
21     while (current != NULL) {
22         if (current->client_id == id &&
23             current->player && current->type == AI) {
24             return current;
25         }
26         current = current->next;
27     }
28     return NULL;
29 }
```

14.167.1.3 get_ppo_buffer()

```
static char* get_ppo_buffer (
    client_t * tmp ) [static]
```

Definition at line 31 of file command_ppo.c.

```
32 {
33     char *buffer = NULL;
34     int size = 0;
35
36     if (!tmp || !tmp->player)
37         return NULL;
38     size = snprintf(NULL, 0, "ppo %d %d %d %d\n", tmp->client_id,
39                     tmp->player->pos_x, tmp->player->pos_y, tmp->player->rotation);
40     buffer = malloc(size + 1);
41     if (!buffer)
42         return NULL;
43     snprintf(buffer, size + 1, "ppo %d %d %d %d\n", tmp->client_id,
44               tmp->player->pos_x, tmp->player->pos_y, tmp->player->rotation);
45     return buffer;
46 }
```

14.167.1.4 send_ppo_command()

```
bool send_ppo_command (
    server_t * server,
    int id )
```

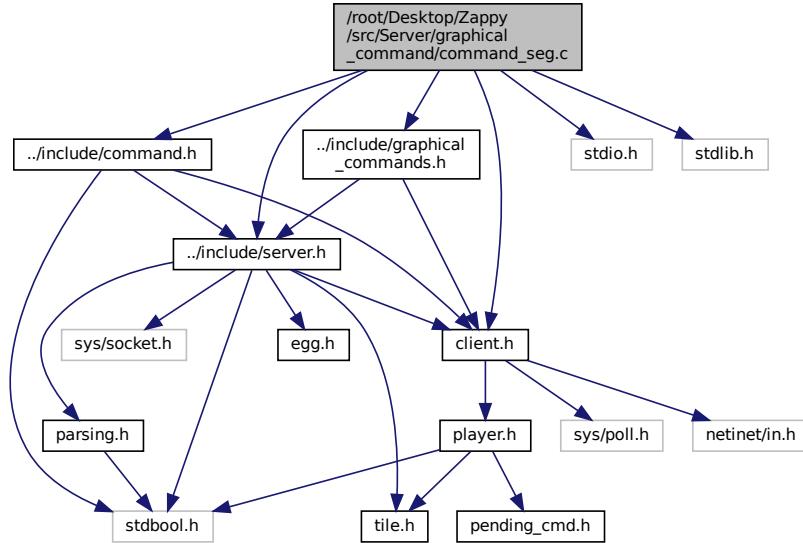
Definition at line 48 of file command_ppo.c.

```
49 {
50     char *buffer = NULL;
51     client_t *tmp = NULL;
52
53     if (!server)
54         return false;
55     tmp = find_client_by_id(server, id);
56     if (!tmp || !tmp->player)
57         return false;
58     buffer = get_ppo_buffer(tmp);
59     for (client_t *cur = server->client; cur != NULL; cur = cur->next) {
60         if (cur->type == GRAPHICAL && cur->is_fully_connected) {
61             write_command_output(cur->client_fd, buffer);
62         }
63     }
64     free(buffer);
65     return true;
66 }
```

14.168 /root/Desktop/Zappy/src/Server/graphical_command/command_seg.c File Reference

```
#include "../include/server.h"
#include "../include/client.h"
#include "../include/command.h"
```

```
#include "../include/graphical_commands.h"
#include <stdio.h>
#include <stdlib.h>
Include dependency graph for command_seg.c:
```



Functions

- void [command_seg \(server_t *server, const char *team_name\)](#)

14.168.1 Function Documentation

14.168.1.1 [command_seg\(\)](#)

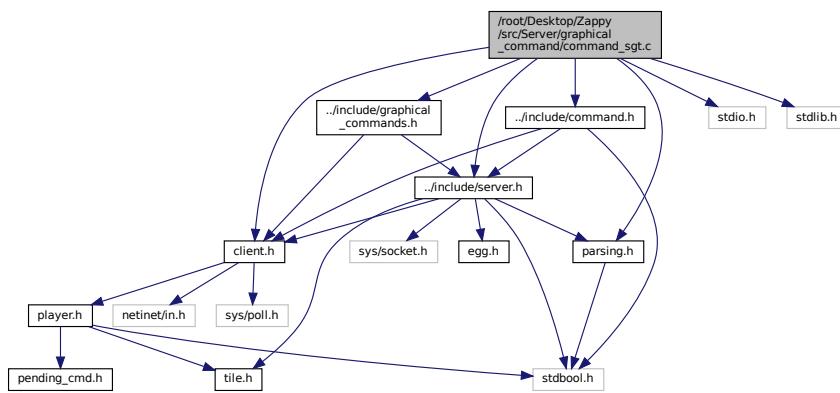
```
void command_seg (
    server_t * server,
    const char * team_name )
```

Definition at line 15 of file command_seg.c.

```
16 {
17     graphical_client_t *current = NULL;
18     char *buffer = NULL;
19     int size = 0;
20
21     if (!server || !server->graphical_clients || !team_name)
22         return;
23     size = snprintf(NULL, 0, "seg %s\n", team_name);
24     buffer = malloc(size + 1);
25     if (!buffer)
26         return;
27     snprintf(buffer, size + 1, "seg %s\n", team_name);
28     current = server->graphical_clients;
29     while (current) {
30         write_command_output(current->client->client_fd, buffer);
31         current = current->next;
32     }
33     free(buffer);
34 }
```

14.169 /root/Desktop/Zappy/src/Server/graphical_command/command_sgt.c File Reference

```
#include "../include/server.h"
#include "../include/client.h"
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include "../include/parsing.h"
#include <stdio.h>
#include <stdlib.h>
Include dependency graph for command_sgt.c:
```



Functions

- void `command_sgt (server_t *server, client_t *client, char **buffer)`

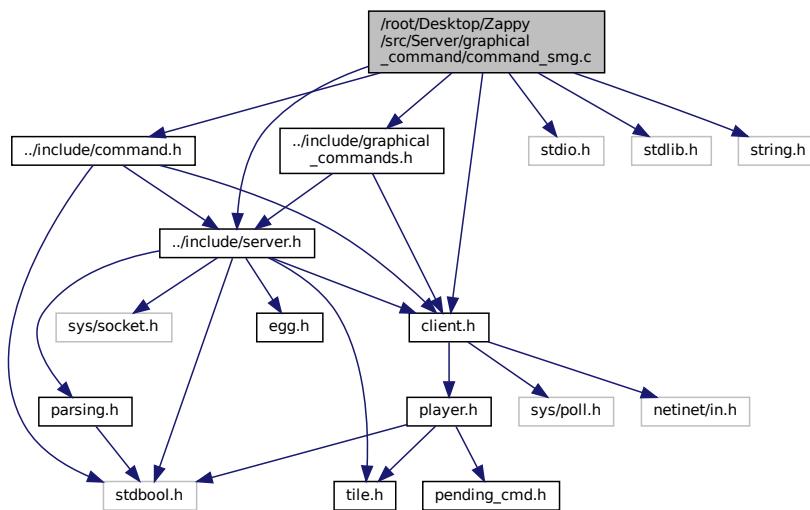
14.169.1 Function Documentation

14.169.1.1 command_sgt()

```
void command_sgt (
    server_t * server,
    client_t * client,
    char ** buffer )
Definition at line 16 of file command_sgt.c.
17 {
18     char *tmp = NULL;
19     int size = 0;
20
21     if (!server || !client || !server->graphical_clients ||
22         arr_len(buffer) != 1)
23         return write_command_output(client->client_fd, "spb\n");
24     tmp = NULL;
25     size = sprintf(NULL, 0, "sgt %d\n", server->parsed_info->frequence);
26     tmp = malloc(size + 1);
27     if (!tmp)
28         return;
29     sprintf(tmp, size + 1, "sgt %d\n", server->parsed_info->frequence);
30     write_command_output(client->client_fd, tmp);
31     free(tmp);
32 }
```

14.170 /root/Desktop/Zappy/src/Server/graphical_command/command_smg.c File Reference

```
#include "../include/server.h"
#include "../include/client.h"
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
Include dependency graph for command_smg.c:
```



Functions

- void [send_smg_command \(server_t *server, const char *msg\)](#)

14.170.1 Function Documentation

14.170.1.1 send_smg_command()

```
void send_smg_command (
    server_t * server,
    const char * msg )
```

Definition at line 16 of file command_smg.c.

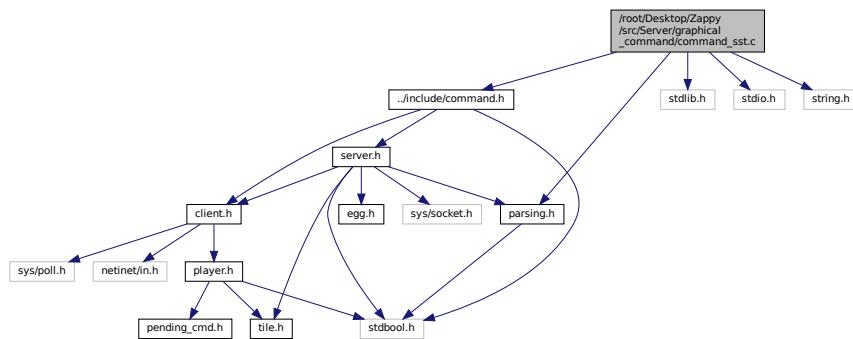
```
17 {
18     graphical_client_t *current = NULL;
19     char *buffer = NULL;
20     int size = 0;
21
22     if (!server || !server->graphical_clients || !msg)
23         return;
24     size = snprintf(NULL, 0, "smg %s\n", msg);
25     buffer = malloc(size + 1);
26     if (!buffer)
27         return;
28     snprintf(buffer, size + 1, "smg %s\n", msg);
29     current = server->graphical_clients;
30     while (current) {
31         write_command_output(current->client->client_fd, buffer);
32         current = current->next;
```

```

33     }
34     free(buffer);
35 }
```

14.171 /root/Desktop/Zappy/src/Server/graphical_command/command_sst.c File Reference

```
#include "../include/command.h"
#include "../include/parsing.h"
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
Include dependency graph for command_sst.c:
```



Functions

- static char * [get_buffer_sst](#) (int time)
- static int [get_time_from_buffer](#) (char *buffer)
- void [command_sst](#) (server_t *server, client_t *client, char **buffer)

14.171.1 Function Documentation

14.171.1.1 [command_sst\(\)](#)

```
void command_sst (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 37 of file command_sst.c.

```

38 {
39     int time;
40     graphical_client_t *graphical_client = NULL;
41     char *tmp_buffer = NULL;
42
43     if (!buffer || client->type != GRAPHICAL || !server->graphical_clients ||
44         arr_len(buffer) != 2)
45         return write_command_output(client->client_fd, "sbp\n");
46     time = get_time_from_buffer(buffer[1]);
47     if (time <= 0)
48         return write_command_output(client->client_fd, "sbp\n");
49     tmp_buffer = get_buffer_sst(time);
50     server->parsed_info->frequence = time;
51     graphical_client = server->graphical_clients;
52     while (graphical_client) {
53         write_command_output(graphical_client->client->client_fd,
54             tmp_buffer);
55         graphical_client = graphical_client->next;
```

```

56     }
57     free(tmp_buffer);
58 }
```

14.171.1.2 get_buffer_sst()

```
static char* get_buffer_sst (
    int time ) [static]
```

Definition at line 14 of file command_sst.c.

```

15 {
16     char *buffer = NULL;
17     int size = snprintf(NULL, 0, "sst %d\n", time);
18
19     buffer = malloc(size + 1);
20     if (!buffer) {
21         return NULL;
22     }
23     snprintf(buffer, size + 1, "sst %d\n", time);
24     return buffer;
25 }
```

14.171.1.3 get_time_from_buffer()

```
static int get_time_from_buffer (
    char * buffer ) [static]
```

Definition at line 27 of file command_sst.c.

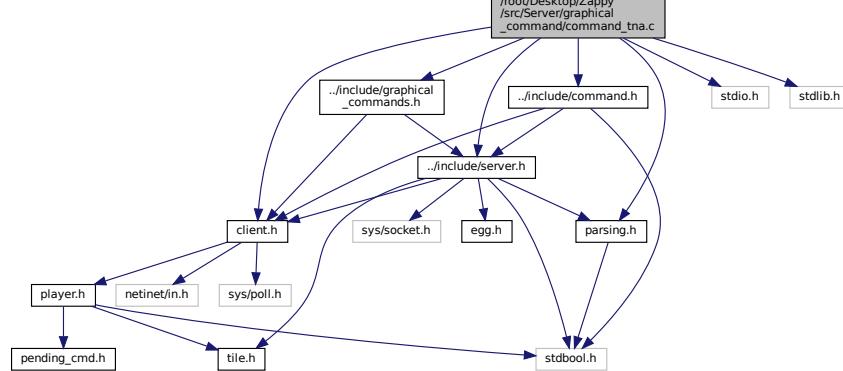
```

28 {
29     int time = 0;
30
31     if (sscanf(buffer, "%d\n", &time) != 1 || time < 0) {
32         return -1;
33     }
34     return time;
35 }
```

14.172 /root/Desktop/Zappy/src/Server/graphical_command/command_tna.c File Reference

```
#include "../include/server.h"
#include "../include/client.h"
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include "../include/parsing.h"
#include <stdio.h>
#include <stdlib.h>
```

Include dependency graph for command_tna.c:



Functions

- static void `send_one_tna_command` (`server_t` *`server`, `client_t` *`client`, const char *`team`)
- void `send_tna_command` (`server_t` *`server`, `client_t` *`client`)
- void `command_tna` (`server_t` *`server`, `client_t` *`client`, char **`buffer`)

14.172.1 Function Documentation

14.172.1.1 `command_tna()`

```
void command_tna (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 42 of file `command_tna.c`.

```
43 {
44     if (!server || !client || !buffer || !server->parsed_info ||
45         !server->parsed_info->names || !server->graphical_clients
46         || arr_len(buffer) != 1)
47         return write_command_output(client->client_fd, "sbp\n");
48     send_tna_command(server, client);
49 }
```

14.172.1.2 `send_one_tna_command()`

```
static void send_one_tna_command (
    server_t * server,
    client_t * client,
    const char * team ) [static]
```

Definition at line 16 of file `command_tna.c`.

```
18 {
19     int size = 0;
20     char *buffer = NULL;
21
22     if (!server || !client || !team)
23         return;
24     size = snprintf(NULL, 0, "tna %s\n", team);
25     buffer = malloc(size + 1);
26     if (!buffer)
27         return;
28     snprintf(buffer, size + 1, "tna %s\n", team);
29     write_command_output(client->client_fd, buffer);
30     free(buffer);
31 }
```

14.172.1.3 `send_tna_command()`

```
void send_tna_command (
    server_t * server,
    client_t * client )
```

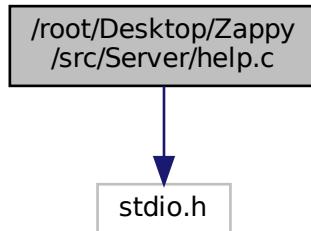
Definition at line 33 of file `command_tna.c`.

```
34 {
35     if (!server || !client)
36         return;
37     for (int i = 0; server->parsed_info->names[i]; i++) {
38         send_one_tna_command(server, client, server->parsed_info->names[i]);
39     }
40 }
```

14.173 /root/Desktop/Zappy/src/Server/help.c File Reference

```
#include <stdio.h>
```

Include dependency graph for help.c:



Functions

- void `display_help` (void)
Display the help message for the server.

14.173.1 Function Documentation

14.173.1.1 `display_help()`

```
void display_help (
    void )
```

Display the help message for the server.

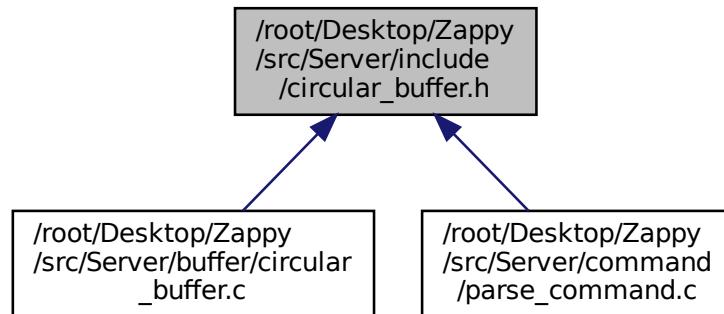
Affiche l'aide d'utilisation du serveur.

Definition at line 13 of file help.c.

```
14 {
15     printf("USAGE: ./zappy_server -p port -x width -y height -n name1 "
16           "name2 ... -c clientsNb -f freq --auto-start on|off --display-eggs "
17           "true|false\n");
18     printf("[v | --verbose]--game_duration time\n");
19     printf("\tport\tis the port number\n");
20     printf("\twidht\tis the width of the world\n");
21     printf("\theight\tis the height of the world\n");
22     printf("\tnameX\tis the name of the team X\n");
23     printf("\tclientsNb\tis the number of authorized clients per team\n");
24     printf("\tfreq\tis the reciprocal of time unit for execution of "
25           "actions\n");
26     printf("\tauto-start\tdoes the greeting is send automatically\n");
27     printf("\tdisplay-eggs\teggs are visible and destructible\n");
28     printf("\tmatch_duration\tis the duration of the match in seconds\n");
29 }
```

14.174 /root/Desktop/Zappy/src/Server/include/circular_buffer.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct `circular_buffer_s`

Macros

- #define `BUFFER_SIZE` 8192

Typedefs

- typedef struct `circular_buffer_s` `circular_buffer_t`

Functions

- void `init_circular_buffer` (`circular_buffer_t` *cb)
- int `add_to_circular_buffer` (`circular_buffer_t` *cb, char byte)
- int `find_command_end` (`circular_buffer_t` *cb)
- char * `extract_command` (`circular_buffer_t` *cb, int cmd_length)

14.174.1 Macro Definition Documentation

14.174.1.1 BUFFER_SIZE

```
#define BUFFER_SIZE 8192
Definition at line 10 of file circular_buffer.h.
```

14.174.2 Typedef Documentation

14.174.2.1 circular_buffer_t

```
typedef struct circular_buffer_s circular_buffer_t
```

14.174.3 Function Documentation

14.174.3.1 add_to_circular_buffer()

```
int add_to_circular_buffer (
    circular_buffer_t * cb,
    char byte )
```

Definition at line 18 of file circular_buffer.c.

```
19 {
20     if (cb->count >= BUFFER_SIZE - 1)
21         return -1;
22     cb->buffer[cb->end] = byte;
23     cb->end = (cb->end + 1) % BUFFER_SIZE;
24     cb->count++;
25     return 0;
26 }
```

14.174.3.2 extract_command()

```
char* extract_command (
    circular_buffer_t * cb,
    int cmd_length )
```

Definition at line 45 of file circular_buffer.c.

```
46 {
47     char *command = malloc(cmd_length + 1);
48
49     if (!command)
50         return NULL;
51     for (int i = 0; i < cmd_length; i++) {
52         command[i] = cb->buffer[cb->start];
53         cb->start = (cb->start + 1) % BUFFER_SIZE;
54         cb->count--;
55     }
56     command[cmd_length] = '\0';
57     return command;
58 }
```

14.174.3.3 find_command_end()

```
int find_command_end (
    circular_buffer_t * cb )
```

Definition at line 28 of file circular_buffer.c.

```
29 {
30     int pos = cb->start;
31
32     if (cb->count < 1)
33         return -1;
34     for (int i = 0; i < cb->count; i++) {
35         if ((cb->buffer[pos] == '\r' && cb->buffer[pos + 1] == '\n')
36             || (cb->buffer[pos] == '\n' && cb->buffer[pos + 1] == '\0'))
37             return -1;
38         if (cb->buffer[pos] == '\n')
39             return i + 1;
40         pos = (pos + 1) % BUFFER_SIZE;
41     }
42     return -1;
43 }
```

14.174.3.4 init_circular_buffer()

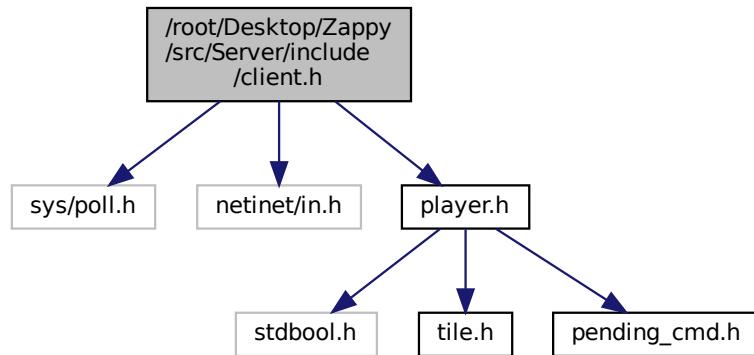
```
void init_circular_buffer (
    circular_buffer_t * cb )
```

Definition at line 11 of file circular_buffer.c.

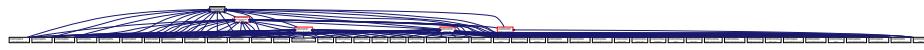
```
12 {
13     cb->start = 0;
14     cb->end = 0;
15     cb->count = 0;
16 }
```

14.175 /root/Desktop/Zappy/src/Server/include/client.h File Reference

```
#include <sys/poll.h>
#include <netinet/in.h>
#include "player.h"
Include dependency graph for client.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [graphical_client_s](#)
- struct [client_s](#)

Typedefs

- typedef struct [graphical_client_s](#) graphical_client_t
- typedef struct [client_s](#) client_t

Enumerations

- enum [client_type_e](#) { GRAPHICAL , AI }

Functions

- void [send_message_to_all_graphic](#) (server_t *server, char *message)
- void [remove_graphic_client](#) (server_t *server, [client_t](#) *client)
- void [add_graphic_client](#) (server_t *server, [client_t](#) *client)
- void [send_map_info_to_one_client](#) (server_t *server, [client_t](#) *client)

14.175.1 TypeDef Documentation

14.175.1.1 client_t

```
typedef struct client_s client_t
```

14.175.1.2 graphical_client_t

```
typedef struct graphical_client_s graphical_client_t
```

14.175.2 Enumeration Type Documentation

14.175.2.1 client_type_e

```
enum client_type_e
```

Enumerator

GRAPHICAL	
AI	

Definition at line 14 of file client.h.

```
14
15     GRAPHICAL,
16     AI
17 };
```

14.175.3 Function Documentation

14.175.3.1 add_graphic_client()

```
void add_graphic_client (
    server_t * server,
    client_t * client )
```

Definition at line 32 of file graphical_client.c.

```
33 {
34     graphical_client_t *new_ref = malloc(sizeof(graphical_client_t));
35
36     if (!new_ref)
37         return;
38     new_ref->client = client;
39     new_ref->next = server->graphical_clients;
40     add_to_graphical_list(server, new_ref);
41 }
```

14.175.3.2 remove_graphic_client()

```
void remove_graphic_client (
    server_t * server,
    client_t * client )
```

Definition at line 59 of file graphical_client.c.

```
60 {
61     graphical_client_t *current = server->graphical_clients;
62     graphical_client_t *prev = NULL;
63
64     if (!server || !client || !server->graphical_clients)
65         return;
66     while (current != NULL) {
67         if (remove_graphical_node(server, client, &current, &prev))
68             return;
69     }
70     remove_fd(server, client->client_fd);
71 }
```

14.175.3.3 send_map_info_to_one_client()

```
void send_map_info_to_one_client (
    server_t * server,
    client_t * client )
```

Definition at line 84 of file graphical_client.c.

```
85 {
86     char **tmp;
87
88     if (!server || !client)
89         return;
90     tmp = malloc(sizeof(char *) * 2);
91     if (!tmp)
92         return;
93     tmp[0] = strdup("sgt\n");
94     tmp[1] = NULL;
95     send_msz_command(server, client);
96     command_sgt(server, client, tmp);
97     send_tile_content_to_one_client(server, client);
98     send_tna_command(server, client);
99     send_all_player_info_to_one_client(server, client);
100    send_enw_command_start(server);
101    free_arr(tmp);
102 }
```

14.175.3.4 send_message_to_all_graphic()

```
void send_message_to_all_graphic (
    server_t * server,
    char * message )
```

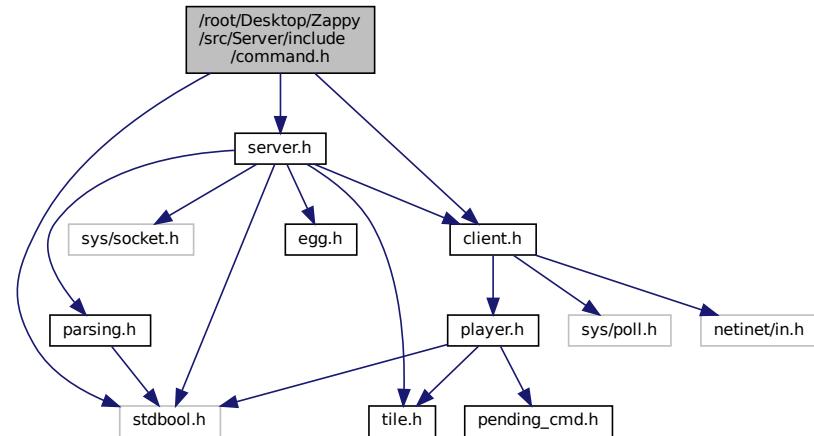
Definition at line 73 of file graphical_client.c.

```
74 {
75     graphical_client_t *current = server->graphical_clients;
76
77     while (current) {
78         if (current->client && current->client->client_fd != -1)
79             write_command_output(current->client->client_fd, message);
80         current = current->next;
81     }
82 }
```

14.176 /root/Desktop/Zappy/src/Server/include/command.h File Reference

```
#include "server.h"
#include <stdbool.h>
#include "client.h"
```

Include dependency graph for command.h:



This graph shows which files directly or indirectly include this file:

Data Structures

- struct [command_data_s](#)

Macros

- #define MAX_LIFE_AFTER_FOOD 126

Typedefs

- typedef struct [command_data_s](#) [command_data_t](#)

Functions

- [resource_type_t determine_type](#) (char *resource_string)
- void [process_next_queued_command](#) ([server_t](#) *server, [client_t](#) *client)
- void [cleanup_player_queue](#) ([player_t](#) *player)
- void [cleanup_client](#) ([client_t](#) *client)
- char * [tile_to_str](#) ([tile_t](#) *tile)
- void [get_message](#) ([server_t](#) *server, [client_t](#) *user)
- void [execute_com](#) ([server_t](#) *server, [client_t](#) *user, char *buffer)
- void [write_command_output](#) (int client_fd, char *msg)
- void [forward](#) ([server_t](#) *server, [client_t](#) *client, char **buffer)
- void [right](#) ([server_t](#) *server, [client_t](#) *client, char **buffer)
- void [left](#) ([server_t](#) *server, [client_t](#) *client, char **buffer)
- void [inventory](#) ([server_t](#) *server, [client_t](#) *client, char **buffer)
- void [look](#) ([server_t](#) *server, [client_t](#) *client, char **buffer)
- void [eject](#) ([server_t](#) *server, [client_t](#) *client, char **buffer)
- void [connect_nbr](#) ([server_t](#) *server, [client_t](#) *client, char **buffer)
- void [take_object](#) ([server_t](#) *server, [client_t](#) *client, char **buffer)
- void [set_object](#) ([server_t](#) *server, [client_t](#) *client, char **buffer)

- void `fork_c (server_t *server, client_t *client, char **buffer)`
- void `broadcast (server_t *server, client_t *user, char **buffer)`
- int `connect_nbr_srv (server_t *server, char *team)`
- bool `can_connect (server_t *server, client_t *user, char *buffer)`
- `client_t * find_client_by_id (server_t *server, int id)`
- void `handle_player_death (server_t *server, client_t *client)`
- bool `check_player_starvation (server_t *server, client_t *client)`
- void `start_incantation (server_t *server, client_t *client, char **buffer)`
- void `finish_incantation (server_t *server, client_t *client)`
- bool `can_start_incantation (server_t *server, client_t *client)`
- `command_data_t get_command_data (void)`
- char * `check_rota_tiles (client_t *user, server_t *server, int i, int j)`
- void `add_pending_cmd (client_t *user, server_t *server, char *buffer, int cmd_index)`
- void `cleanup_pending (player_t *player)`
- void `command_msz (server_t *server, client_t *client, char **buffer)`
- void `command_bct (server_t *server, client_t *client, char **buffer)`
- void `command_mtc (server_t *server, client_t *client, char **buffer)`
- void `command_tna (server_t *server, client_t *client, char **buffer)`
- void `command_ppo (server_t *server, client_t *client, char **buffer)`
- void `command_plv (server_t *server, client_t *client, char **buffer)`
- void `command_pie (server_t *server, int x, int y, int result)`
- void `command_pex (server_t *server, client_t *client)`
- void `command_pin (server_t *server, client_t *client, char **buffer)`
- void `command_pkf (server_t *server, client_t *client)`
- void `command_pdr (server_t *server, client_t *client, resource_type_t resource_type)`
- void `command_pgt (server_t *server, client_t *client, resource_type_t resource_type)`
- void `send_enw_command (server_t *server, client_t *client, int egg_id)`
- void `command_sst (server_t *server, client_t *client, char **buffer)`
- void `command_pdi (server_t *server, client_t *client)`
- void `send_enw_command_start (server_t *server)`
- void `send_mtc_to_all_graphical_clients (server_t *server)`
- void `send_bct_to_all_graphical_clients (server_t *server, int x, int y)`
- void `send_pnw_command_to_all (server_t *server, client_t *client)`
- char ** `str_to_word_arr (char *str, char *delim)`

14.176.1 Macro Definition Documentation

14.176.1.1 MAX_LIFE_AFTER_FOOD

```
#define MAX_LIFE_AFTER_FOOD 126
Definition at line 10 of file command.h.
```

14.176.2 Typedef Documentation

14.176.2.1 command_data_t

```
typedef struct command_data_s command_data_t
```

14.176.3 Function Documentation

14.176.3.1 add_pending_cmd()

```
void add_pending_cmd (
    client_t * user,
    server_t * server,
    char * buffer,
    int cmd_index )
```

Definition at line 28 of file parse_command_utils.c.

```
30 {
31     command_data_t data = get_command_data();
32     char **tmp = NULL;
33
34     if (cmd_index == 9) {
35         tmp = str_to_word_arr(buffer, " ");
36         start_incantation(server, user, tmp);
37         return free_arr(tmp);
38     }
39     if (cmd_index == 10)
40         command_pfk(server, user);
41     user->player->pending_cmd->args = strdup(buffer);
42     user->player->pending_cmd->func = data.functions[cmd_index];
43     if (data.times[cmd_index] > 0)
44         user->player->busy_until =
45             server->current_tick + data.times[cmd_index];
46 }
```

14.176.3.2 broadcast()

```
void broadcast (
    server_t * server,
    client_t * user,
    char ** buffer )
```

Definition at line 149 of file broadcast.c.

```
150 {
151     char *message;
152     client_t *current;
153
154     if (!user || !user->player || !buffer || arr_len(buffer) < 2)
155         return write_command_output(user->client_fd, "ko\n");
156     message = build_broadcast_message(buffer);
157     if (!message)
158         return write_command_output(user->client_fd, "ko\n");
159     current = server->client;
160     if (current)
161         current = current->next;
162     command_pbc(server, user, message);
163     while (current) {
164         if (current->player && current != user && current->type != GRAPHICAL)
165             send_broadcast_to_client(server, user, current, message);
166         current = current->next;
167     }
168     free(message);
169     write_command_output(user->client_fd, "ok\n");
170 }
```

14.176.3.3 can_connect()

```
bool can_connect (
    server_t * server,
    client_t * user,
    char * buffer )
```

Definition at line 38 of file parse_command_utils_bis.c.

```
39 {
40     if (!is_valid_team_name(buffer, server, user)){
41         write_command_output(user->client_fd, "ko\n");
42         return false;
43     }
44     if (user->type != GRAPHICAL &&
45         connect_nbr_srv(server, user->player->team_name) < 0) {
46         if (user->player->team_name) {
47             free(user->player->team_name);
48             user->player->team_name = NULL;
49         }
50         write_command_output(user->client_fd, "ko\n");
```

```

51         return false;
52     }
53     return true;
54 }
```

14.176.3.4 can_start_incantation()

```
bool can_start_incantation (
    server_t * server,
    client_t * client )
```

Definition at line 91 of file start_incantation.c.

```

92 {
93     int required_players = 0;
94     int prerequisite_level = client->player->level + 1;
95     tile_t *tile = &server->map[client->player->pos_y][client->player->pos_x];
96     int current_players = 0;
97
98     if (prerequisite_level > 7 ||
99         !has_enough_resources(tile, prerequisite_level))
100        return false;
101     required_players = how_many_players_needed(prerequisite_level);
102     current_players =
103         nb_valid_players_on_tile(server, tile, client->player->level);
104     if (current_players < required_players)
105        return false;
106     return true;
107 }
```

14.176.3.5 check_player_starvation()

```
bool check_player_starvation (
    server_t * server,
    client_t * client )
```

Definition at line 39 of file player_death.c.

```

40 {
41     int food_amount = 0;
42     bool dead = false;
43
44     if (!client || !client->player || !client->is_fully_connected)
45        return dead;
46     client->player->life--;
47     if (client->player->life <= 0) {
48         food_amount = how_many_in_inventory(client->player, FOOD);
49         if (food_amount > 0) {
50             remove_item_from_inventory(client->player, FOOD, 1);
51             client->player->life = MAX_LIFE_AFTER_FOOD;
52         } else {
53             handle_player_death(server, client);
54             dead = true;
55         }
56     }
57     return dead;
58 }
```

14.176.3.6 check_rota_tiles()

```
char* check_rota_tiles (
    client_t * user,
    server_t * server,
    int i,
    int j)
```

Definition at line 47 of file look_utils.c.

```

48 {
49     switch (user->player->rotation) {
50     case UP:
51         return check_up(user, i, j, server);
52     case DOWN:
53         return check_down(user, i, j, server);
54     case LEFT:
55         return check_left(user, i, j, server);
56     case RIGHT:
57         return check_right(user, i, j, server);
```

```

58     default:
59         return strdup("");
60     }
61 }
```

14.176.3.7 cleanup_client()

```
void cleanup_client (
    client_t * client )
```

Definition at line 63 of file parse_command_utils.c.

```

64 {
65     if (!client)
66         return;
67     if (client->player) {
68         cleanup_player_queue(client->player);
69         cleanup_pending(client->player);
70         free(client->player);
71         client->player = NULL;
72     }
73 }
```

14.176.3.8 cleanup_pending()

```
void cleanup_pending (
    player_t * player )
```

Definition at line 16 of file parse_command_utils.c.

```

17 {
18     if (!player || !player->pending_cmd)
19         return;
20     if (player->pending_cmd->args){
21         free(player->pending_cmd->args);
22         player->pending_cmd->args = NULL;
23     }
24     free(player->pending_cmd);
25     player->pending_cmd = NULL;
26 }
```

14.176.3.9 cleanup_player_queue()

```
void cleanup_player_queue (
    player_t * player )
```

Definition at line 48 of file parse_command_utils.c.

```

49 {
50     if (!player || !player->command_queue)
51         return;
52     for (int i = 0; i < player->queue_size; i++) {
53         if (player->command_queue[i]) {
54             free(player->command_queue[i]);
55             player->command_queue[i] = NULL;
56         }
57     }
58     free(player->command_queue);
59     player->command_queue = NULL;
60     player->queue_size = 0;
61 }
```

14.176.3.10 command_bct()

```
void command_bct (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 99 of file command_bct.c.

```

100 {
101     int x = 0;
102     int y = 0;
103
104     if (!server || !client || !buffer || !server->graphical_clients ||
105         arr_len(buffer) != 3 || sscanf(buffer[1], "%d", &x) != 1 ||
```

```

106         sscanf(buffer[2], "%d", &y) != 1 ||
107         x < 0 || y < 0 ||
108         y >= server->parsed_info->height ||
109         x >= server->parsed_info->width);
110     return write_command_output(client->client_fd, "sbp\n");
111     send_bct_command(server, client, x, y);
112 }

```

14.176.3.11 command_msz()

```

void command_msz (
    server_t * server,
    client_t * client,
    char ** buffer )

```

Definition at line 35 of file command_msz.c.

```

36 {
37     if (!server || !client || !server->graphical_clients ||
38         arr_len(buffer) != 1)
39     return write_command_output(client->client_fd, "sbp\n");
40     send_msz_command(server, client);
41 }

```

14.176.3.12 command_mtc()

```

void command_mtc (
    server_t * server,
    client_t * client,
    char ** buffer )

```

Definition at line 114 of file command_bct.c.

```

115 {
116     if (!server || !client || !server->graphical_clients
117         || arr_len(buffer) != 1)
118     return write_command_output(client->client_fd, "sbp\n");
119     send_tile_content_to_one_client(server, client);
120 }

```

14.176.3.13 command_pdi()

```

void command_pdi (
    server_t * server,
    client_t * client )

```

Definition at line 13 of file command_pdi.c.

```

14 {
15     graphical_client_t *graphical_client = NULL;
16     char *buffer = NULL;
17     int size = 0;
18
19     if (!client || !client->player || !server->graphical_clients)
20         return;
21     size = sprintf(NULL, 0, "pdi #%d\n", client->client_id);
22     buffer = malloc(size + 1);
23     if (buffer == NULL)
24         return;
25     sprintf(buffer, "pdi #%d\n", client->client_id);
26     graphical_client = server->graphical_clients;
27     while (graphical_client != NULL) {
28         write_command_output(graphical_client->client->client_fd,
29             buffer);
30         graphical_client = graphical_client->next;
31     }
32     free(buffer);
33 }

```

14.176.3.14 command_pdr()

```

void command_pdr (
    server_t * server,

```

```
    client_t * client,
    resource_type_t resource_type )
```

Definition at line 12 of file command_pdr.c.

```
14 {
15     graphical_client_t *graphical_client = NULL;
16     char *buffer = NULL;
17     int size = 0;
18
19     if (!client || !client->player || !server->graphical_clients)
20         return;
21     size = sprintf(NULL, 0, "pdr #%d %d\n", client->client_id, resource_type);
22     buffer = malloc(size + 1);
23     if (buffer == NULL)
24         return write_command_output(client->client_fd, "ko\n");
25     sprintf(buffer, "pdr #%d %d\n", client->client_id, resource_type);
26     graphical_client = server->graphical_clients;
27     while (graphical_client != NULL) {
28         write_command_output(graphical_client->client->client_fd, buffer);
29         graphical_client = graphical_client->next;
30     }
31     free(buffer);
32 }
```

14.176.3.15 command_pex()

```
void command_pex (
    server_t * server,
    client_t * client )
```

Definition at line 12 of file command_pex.c.

```
13 {
14     graphical_client_t *graphical_client = NULL;
15     char *buffer = NULL;
16     int size = 0;
17
18     if (!client || !client->player || !server->graphical_clients)
19         return;
20     size = sprintf(NULL, 0, "pex #%d\n", client->client_id);
21     buffer = malloc(size + 1);
22     if (buffer == NULL)
23         return write_command_output(client->client_fd, "ko\n");
24     sprintf(buffer, "pex #%d\n", client->client_id);
25     graphical_client = server->graphical_clients;
26     while (graphical_client != NULL) {
27         write_command_output(graphical_client->client->client_fd,
28             buffer);
29         graphical_client = graphical_client->next;
30     }
31     free(buffer);
32 }
```

14.176.3.16 command_pfk()

```
void command_pfk (
    server_t * server,
    client_t * client )
```

Definition at line 12 of file command_pfk.c.

```
13 {
14     graphical_client_t *graphical_client = NULL;
15     char *buffer = NULL;
16     int size = 0;
17
18     if (!client || !client->player || !server->graphical_clients)
19         return;
20     size = sprintf(NULL, 0, "pfk #%d\n", client->client_id);
21     buffer = malloc(size + 1);
22     if (buffer == NULL)
23         return write_command_output(client->client_fd, "ko\n");
24     sprintf(buffer, "pfk #%d\n", client->client_id);
25     graphical_client = server->graphical_clients;
26     while (graphical_client != NULL) {
27         write_command_output(graphical_client->client->client_fd, buffer);
28         graphical_client = graphical_client->next;
29     }
30     free(buffer);
31 }
```

14.176.3.17 command_pgt()

```
void command_pgt (
    server_t * server,
    client_t * client,
    resource_type_t resource_type )
```

Definition at line 12 of file command_pgt.c.

```
14 {
15     graphical_client_t *graphical_client = NULL;
16     char *buffer = NULL;
17     int size = 0;
18
19     if (!client || !client->player || !server->graphical_clients)
20         return;
21     size = snprintf(NULL, 0, "pgt #%d %d\n", client->client_id, resource_type);
22     buffer = malloc(size + 1);
23     if (buffer == NULL)
24         return;
25     sprintf(buffer, "pgt #%d %d\n", client->client_id, resource_type);
26     graphical_client = server->graphical_clients;
27     while (graphical_client != NULL) {
28         write_command_output(graphical_client->client_fd, buffer);
29         graphical_client = graphical_client->next;
30     }
31     free(buffer);
32 }
```

14.176.3.18 command_pie()

```
void command_pie (
    server_t * server,
    int x,
    int y,
    int result )
```

Definition at line 28 of file command_pie.c.

```
29 {
30     client_t *current = NULL;
31     char *buffer = NULL;
32
33     if (!server || !server->graphical_clients)
34         return;
35     buffer = get_pie_buffer(x, y, result);
36     if (!buffer)
37         return;
38     current = server->client;
39     while (current) {
40         write_command_output(current->client_fd, buffer);
41         current = current->next;
42     }
43     free(buffer);
44 }
```

14.176.3.19 command_pin()

```
void command_pin (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 78 of file command_pin.c.

```
79 {
80     client_t *recipient = NULL;
81     int id = -1;
82
83     if (!server || !client || !server->graphical_clients ||
84         arr_len(buffer) != 2 ||
85         sscanf(buffer[1], "#%d\n", &id) != 1 || id < 0)
86         return write_command_output(client->client_fd, "sbp\n");
87     recipient = find_client_by_id(server, id);
88     if (!recipient || recipient->type != AI) {
89         write_command_output(client->client_fd, "sbp\n");
90         return;
91     }
92     send_pin_command(server, recipient, client);
93 }
```

14.176.3.20 command_plv()

```
void command_plv (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 51 of file command_plv.c.

```
52 {
53     client_t *recipient = NULL;
54     int id = -1;
55
56     if (!server || !client || !buffer ||
57         !server->graphical_clients || arr_len(buffer) != 2 ||
58         sscanf(buffer[1], "#%d\n", &id) != 1)
59         return write_command_output(client->client_fd, "sbp\n");
60     recipient = find_client_by_id(server, id);
61     if (!recipient || recipient->type != AI) {
62         write_command_output(client->client_fd, "sbp\n");
63         return;
64     }
65     send_plv_command(server, recipient, client);
66 }
```

14.176.3.21 command_ppo()

```
void command_ppo (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 68 of file command_ppo.c.

```
69 {
70     int id = 0;
71
72     if (!server || !client || !buffer || !server->graphical_clients ||
73         arr_len(buffer) != 2 || sscanf(buffer[1], "#%d\n", &id) != 1 ||
74         id < 0 || !find_client_by_id(server, id) ||
75         !send_ppo_command(server, id)) {
76         return write_command_output(client->client_fd, "sbp\n");
77     }
78 }
```

14.176.3.22 command_sst()

```
void command_sst (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 37 of file command_sst.c.

```
38 {
39     int time;
40     graphical_client_t *graphical_client = NULL;
41     char *tmp_buffer = NULL;
42
43     if (!buffer || client->type != GRAPHICAL || !server->graphical_clients ||
44         arr_len(buffer) != 2)
45         return write_command_output(client->client_fd, "sbp\n");
46     time = get_time_from_buffer(buffer[1]);
47     if (time <= 0)
48         return write_command_output(client->client_fd, "sbp\n");
49     tmp_buffer = get_buffer_sst(time);
50     server->parsed_info->frequency = time;
51     graphical_client = server->graphical_clients;
52     while (graphical_client) {
53         write_command_output(graphical_client->client->client_fd,
54             tmp_buffer);
55         graphical_client = graphical_client->next;
56     }
57     free(tmp_buffer);
58 }
```

14.176.3.23 command_tna()

```
void command_tna (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 42 of file command_tna.c.

```
43 {
44     if (!server || !client || !buffer || !server->parsed_info ||
45         !server->parsed_info->names || !server->graphical_clients
46         || arr_len(buffer) != 1)
47         return write_command_output(client->client_fd, "sbp\n");
48     send_tna_command(server, client);
49 }
```

14.176.3.24 connect_nbr()

```
void connect_nbr (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 64 of file connect_nbr.c.

```
65 {
66     int available_slots;
67
68     if (!client || !client->player || !client->player->team_name ||
69         arr_len(buffer) != 1) {
70         write_command_output(client->client_fd, "ko\n");
71         return;
72     }
73     available_slots = connect_nbr_srv(server, client->player->team_name);
74     format_response(available_slots, client);
75 }
```

14.176.3.25 connect_nbr_srv()

```
int connect_nbr_srv (
    server_t * server,
    char * team )
```

Definition at line 52 of file connect_nbr.c.

```
53 {
54     int team_eggs = count_team_eggs(server, team);
55     int team_players = count_team_players(server, team);
56     int max_clients = server->parsed_info->client_nb;
57     int available_slots = max_clients - (team_players + team_eggs);
58
59     if (available_slots < 0)
60         available_slots = 0;
61     return available_slots;
62 }
```

14.176.3.26 determine_type()

```
resource_type_t determine_type (
    char * resource_string )
```

Definition at line 16 of file take_object.c.

```
17 {
18     if (strcmp(resource_string, "food", 4) == 0)
19         return FOOD;
20     if (strcmp(resource_string, "linemate", 8) == 0)
21         return LINEMATE;
22     if (strcmp(resource_string, "deraumere", 9) == 0)
23         return DERAUMERE;
24     if (strcmp(resource_string, "sibur", 5) == 0)
25         return SIBUR;
26     if (strcmp(resource_string, "mendiane", 8) == 0)
27         return MENDIANE;
28     if (strcmp(resource_string, "phiras", 6) == 0)
29         return PHIRAS;
30     if (strcmp(resource_string, "thystame", 8) == 0)
```

```

31     return THYSTAME;
32     return COUNT;
33 }

```

14.176.3.27 eject()

```

void eject (
    server_t * server,
    client_t * client,
    char ** buffer )

```

Definition at line 124 of file eject.c.

```

125 {
126     float x = 0;
127     float y = 0;
128
129     if (!client || !client->player || arr_len(buffer) != 1)
130         return write_command_output(client->client_fd, "ko\n");
131     convert_rotation_to_vector(client, &x, &y);
132     push_client(server, client, x, y);
133     push_eggs(server, client->player->pos_x, client->player->pos_y);
134     command_pex(server, client);
135     write_command_output(client->client_fd, "ok\n");
136 }

```

14.176.3.28 execute_com()

```

void execute_com (
    server_t * server,
    client_t * user,
    char * buffer )

```

Definition at line 115 of file parse_command.c.

```

116 {
117     if (!user)
118         return;
119     if (!user->is_fully_connected && can_connect(server, user, buffer)){
120         user->is_fully_connected = true;
121         if (user->type == GRAPHICAL) {
122             add_graphic_client(server, user);
123             send_map_info_to_one_client(server, user);
124             return;
125         } else
126             return send_info_new_client(server, user);
127     }
128     if (!find_and_execute(server, user, buffer)){
129         if (user->type == GRAPHICAL)
130             return write_command_output(user->client_fd, "suc\n");
131         write_command_output(user->client_fd, "ko\n");
132     }
133 }

```

14.176.3.29 find_client_by_id()

```

client_t* find_client_by_id (
    server_t * server,
    int id )

```

Definition at line 17 of file command_ppo.c.

```

18 {
19     client_t *current = server->client;
20
21     while (current != NULL) {
22         if (current->client_id == id &&
23             current->player && current->type == AI) {
24             return current;
25         }
26         current = current->next;
27     }
28     return NULL;
29 }

```

14.176.3.30 finish_incantation()

```
void finish_incantation (
    server_t * server,
    client_t * client )
```

Definition at line 81 of file finish_incantation.c.

```
82 {
83     tile_t *tile = &server->map[client->player->pos_y][client->player->pos_x];
84     int old_level;
85
86     if (!client || !client->player || !client->player->is_in_incantation ||
87         client->player->busy_until > server->current_tick)
88         return;
89     old_level = client->player->level;
90     client->player->is_in_incantation = false;
91     if (!can_start_incantation(server, client)) {
92         command_pie(server, client->player->pos_x,
93                     client->player->pos_y, 0);
94         return handle_incantation_failure(client);
95     }
96     handle_incantation_success(client, tile, old_level, server);
97     command_pie(server, client->player->pos_x,
98                 client->player->pos_y, 1);
99     send_plv_to_all(server, client);
100 }
```

14.176.3.31 fork_c()

```
void fork_c (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 27 of file fork.c.

```
28 {
29     egg_t *new_egg;
30     int egg_id;
31
32     if (!client || !client->player || arr_len(buffer) != 1) {
33         write_command_output(client->client_fd, "ko\n");
34         return;
35     }
36     egg_id = get_next_egg_id(server);
37     new_egg = create_egg(egg_id, client->player->pos_x, client->player->pos_y,
38                         client->player->team_name);
39     if (!new_egg) {
40         write_command_output(client->client_fd, "ko\n");
41         return;
42     }
43     add_egg(server, new_egg);
44     send_enw_command(server, client, egg_id);
45     write_command_output(client->client_fd, "ok\n");
46 }
```

14.176.3.32 forward()

```
void forward (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 44 of file forward.c.

```
45 {
46     if (!client || !client->player || arr_len(buffer) != 1) {
47         write_command_output(client->client_fd, "ko\n");
48         return;
49     }
50     if (client->player->rotation != UP && client->player->rotation != DOWN
51         && client->player->rotation != LEFT
52         && client->player->rotation != RIGHT) {
53         perror("Unexpected forward rotation");
54         write_command_output(client->client_fd, "ko\n");
55         return;
56     }
57     change_map_pos(server, client);
58     send_ppo_command(server, client->client_id);
59     write_command_output(client->client_fd, "ok\n");
```

```
60 }
```

14.176.3.33 get_command_data()

```
command_data_t get_command_data (
    void )
```

Definition at line 31 of file parse_command.c.

```
32 {
33     static const char *comm_char[] = {"Forward", "Right", "Left",
34         "Inventory", "Look", "Eject", "Connect_nbr", "Take", "Set",
35         "Incantation", "Fork", "Broadcast", "msz", "bct", "mtc",
36         "tna", "ppo", "plv", "pin", "sgt", "sst", NULL};
37     static void (*comm_func[]) (server_t *, client_t *, char **) =
38         {forward, right, left, inventory, look, eject,
39         connect_nbr, take_object, set_object, start_incantation,
40         fork_c, broadcast, command_msz, command_bct, command_mtc,
41         command_tna, command_ppo, command_plv, command_pin,
42         command_sgt, command_sst, NULL};
43     static int comm_times[] = {7, 7, 7, 1, 7, 7, 0, 7, 7, 300, 42, 7, 0,
44         0, 0, 0, 0, 0, 0, 0};
45     static enum client_type_e accepted_types[] = {AI, AI, AI, AI, AI,
46         AI, AI, AI, AI, AI, AI, GRAPHICAL, GRAPHICAL, GRAPHICAL,
47         GRAPHICAL, GRAPHICAL, GRAPHICAL, GRAPHICAL, GRAPHICAL};
48     command_data_t data = {comm_char, comm_func, comm_times, accepted_types};
49
50     return data;
51 }
```

14.176.3.34 get_message()

```
void get_message (
    server_t * server,
    client_t * user )
```

Definition at line 147 of file parse_command.c.

```
148 {
149     circular_buffer_t temp_buffer;
150     char byte;
151     int bytes_read;
152     int cmd_length;
153
154     init_circular_buffer(&temp_buffer);
155     while (1) {
156         bytes_read = read(user->client_poll->fd, &byte, 1);
157         if (check_disconnect(bytes_read, user, server) == 1)
158             return;
159         if (add_to_circular_buffer(&temp_buffer, byte) == -1) {
160             write_command_output(user->client_fd, "ko\n");
161             return;
162         }
163         cmd_length = find_command_end(&temp_buffer);
164         if (cmd_length > 0) {
165             check_command(&temp_buffer, cmd_length, server, user);
166             break;
167         }
168     }
169 }
```

14.176.3.35 handle_player_death()

```
void handle_player_death (
    server_t * server,
    client_t * client )
```

Definition at line 15 of file player_death.c.

```
16 {
17     tile_t *tile;
18
19     if (!client || !client->player)
20         return;
21     write_command_output(client->client_fd, "dead\n");
22     command_pdi(server, client);
23     if (server->map && client->player->pos_y < server->parsed_info->height &&
24         client->player->pos_x < server->parsed_info->width) {
25         tile = &server->map[client->player->pos_y][client->player->pos_x];
```

```

26     tile_remove_player(tile, client->client_id);
27 }
28 if (client->player) {
29     free_inventory(client->player);
30     cleanup_player_queue(client->player);
31     cleanup_pending(client->player);
32     free(client->player->team_name);
33     free(client->player);
34     client->player = NULL;
35 }
36 remove_fd(server, client->client_fd);
37 }

```

14.176.3.36 inventory()

```

void inventory (
    server_t * server,
    client_t * client,
    char ** buffer )

```

Definition at line 14 of file inventory.c.

```

15 {
16     char *content;
17
18     if (!server || !client || !client->player || arr_len(buffer) != 1)
19         return write_command_output(client->client_fd, "ko\n");
20     content = get_inventory_content(client->player);
21     write_command_output(client->client_fd, content);
22     free(content);
23 }

```

14.176.3.37 left()

```

void left (
    server_t * server,
    client_t * client,
    char ** buffer )

```

Definition at line 32 of file left.c.

```

33 {
34     if (!server || !client || !client->player || arr_len(buffer) != 1)
35         return write_command_output(client->client_fd, "ko\n");
36     if (client->player->rotation != UP && client->player->rotation != DOWN
37         && client->player->rotation != LEFT
38         && client->player->rotation != RIGHT) {
39         perror("Unexpected left rotation");
40         return write_command_output(client->client_fd, "ko\n");
41     }
42     change_rot(client);
43     send_ppo_command(server, client->client_id);
44     write_command_output(client->client_fd, "ok\n");
45 }

```

14.176.3.38 look()

```

void look (
    server_t * server,
    client_t * client,
    char ** buffer )

```

Definition at line 105 of file look.c.

```

106 {
107     char **level_tiles = NULL;
108     size_t total_len = 0;
109     char *res;
110
111     if (!server || !user || !user->player || arr_len(buffer) != 1)
112         return write_command_output(user->client_fd, "ko\n");
113     total_len = get_total_size(&level_tiles, user, server);
114     if (!level_tiles)
115         return write_command_output(user->client_fd, "ko\n");
116     res = format_look(total_len, level_tiles, user);
117     free(level_tiles);
118     write_command_output(user->client_fd, res);

```

```
119     free(res);
120 }
```

14.176.3.39 process_next_queued_command()

```
void process_next_queued_command (
    server_t * server,
    client_t * client )
```

Definition at line 33 of file queue_connection.c.

```
34 {
35     char *next_command;
36     bool was_full;
37
38     if (!client || !client->player || !server ||
39         client->player->queue_size <= 0 || !client->player->command_queue)
40         return;
41     was_full = (client->player->queue_size >= 10);
42     next_command = client->player->command_queue[0];
43     for (int i = 0; i < client->player->queue_size - 1; i++) {
44         client->player->command_queue[i] =
45             client->player->command_queue[i + 1];
46     }
47     client->player->command_queue[client->player->queue_size - 1] = NULL;
48     client->player->queue_size--;
49     check_rebuild(was_full, client, server);
50     if (next_command) {
51         execute_com(server, client, next_command);
52         free(next_command);
53         next_command = NULL;
54     }
55 }
```

14.176.3.40 right()

```
void right (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 32 of file right.c.

```
33 {
34     if (!server || !client || !client->player || arr_len(buffer) != 1)
35         return write_command_output(client->client_fd, "ko\n");
36     if (client->player->rotation != UP && client->player->rotation != DOWN
37         && client->player->rotation != LEFT
38         && client->player->rotation != RIGHT) {
39         perror("Unexpected right rotation");
40         return write_command_output(client->client_fd, "ko\n");
41     }
42     change_rot(client);
43     send_ppo_command(server, client->client_id);
44     write_command_output(client->client_fd, "ok\n");
45 }
```

14.176.3.41 send_bct_to_all_graphical_clients()

```
void send_bct_to_all_graphical_clients (
    server_t * server,
    int x,
    int y )
```

Definition at line 64 of file command_bct.c.

```
65 {
66     graphical_client_t *current = server->graphical_clients;
67
68     if (!server || !server->graphical_clients)
69         return;
70     while (current) {
71         send_bct_command(server, current->client, x, y);
72         current = current->next;
73     }
74 }
```

14.176.3.42 send_enw_command()

```
void send_enw_command (
    server_t * server,
    client_t * client,
    int egg_id )
```

Definition at line 28 of file command_enw.c.

```
29 {
30     char *buffer = NULL;
31     graphical_client_t *graphical_client = NULL;
32
33     if (!server || !client || !client->player || !server->graphical_clients)
34         return;
35     buffer = get_buffer_for_enw(egg_id, client->client_id,
36                                client->player->pos_x, client->player->pos_y);
37     if (!buffer)
38         return;
39     graphical_client = server->graphical_clients;
40     while (graphical_client) {
41         write_command_output(graphical_client->client->client_fd, buffer);
42         graphical_client = graphical_client->next;
43     }
44     free(buffer);
45 }
```

14.176.3.43 send_enw_command_start()

```
void send_enw_command_start (
    server_t * server )
```

Definition at line 60 of file command_enw.c.

```
61 {
62     graphical_client_t *graphical_client;
63     egg_t *egg;
64
65     if (!server || !server->graphical_clients)
66         return;
67     egg = server->eggs;
68     while (egg) {
69         graphical_client = server->graphical_clients;
70         while (graphical_client) {
71             send_enw_command_to_client(graphical_client->client, egg);
72             graphical_client = graphical_client->next;
73         }
74         egg = egg->next;
75     }
76 }
```

14.176.3.44 send_mtc_to_all_graphical_clients()

```
void send_mtc_to_all_graphical_clients (
    server_t * server )
```

Definition at line 76 of file command_bct.c.

```
77 {
78     graphical_client_t *current = server->graphical_clients;
79
80     if (!server || !server->graphical_clients)
81         return;
82     while (current) {
83         send_tile_content_to_one_client(server, current->client);
84         current = current->next;
85     }
86 }
```

14.176.3.45 send_pnw_command_to_all()

```
void send_pnw_command_to_all (
    server_t * server,
    client_t * client )
```

Definition at line 50 of file command_pnw.c.

```
51 {
52     client_t *current = NULL;
```

```

53     if (!server || !client || !client->player)
54         return;
55     current = server->client;
56     while (current) {
57         if (current->type == GRAPHICAL && current->is_fully_connected)
58             send_pnw_command(server, client, current);
59         current = current->next;
60     }
61 }
62 }
```

14.176.3.46 set_object()

```

void set_object (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 14 of file set_object.c.

```

15 {
16     resource_type_t resource_type;
17
18     if (!client || !client->player || arr_len(buffer) != 2)
19         return write_command_output(client->client_fd, "ko\n");
20     resource_type = determine_type(buffer[1]);
21     if (resource_type == COUNT)
22         return write_command_output(client->client_fd, "ko\n");
23     if (how_many_in_inventory(client->player, resource_type) <= 0)
24         return write_command_output(client->client_fd, "ko\n");
25     remove_item_from_inventory(client->player, resource_type, 1);
26     server->map[client->player->pos_y]
27         [client->player->pos_x].resources[resource_type]++;
28     server->current_resources[resource_type]++;
29     command_pdr(server, client, resource_type);
30     send_pin_to_all(server, client);
31     send_bct_to_all_graphical_clients(server, client->player->pos_x,
32         client->player->pos_y);
33     write_command_output(client->client_fd, "ok\n");
34 }
```

14.176.3.47 start_incantation()

```

void start_incantation (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 109 of file start_incantation.c.

```

110 {
111     if (!server || !client || !client->player || arr_len(buffer) != 1)
112         return write_command_output(client->client_fd, "ko\n");
113     if (!can_start_incantation(server, client))
114         return write_command_output(client->client_fd, "ko\n");
115     command_pic(server, client->player->pos_x, client->player->pos_y,
116         client->player->level);
117     set_busy_all(server, client, 300);
118     client->player->is_in_incantation = true;
119     client->player->incantation_leader_id = client->client_id;
120     write_command_output(client->client_fd, "Elevation underway\n");
121 }
```

14.176.3.48 str_to_word_arr()

```

char** str_to_word_arr (
    char * str,
    char * delim )
```

Definition at line 146 of file array_function.c.

```

147 {
148     char *str_copy;
149     char **result;
150     int len = 0;
151
152     if (!str || !delim)
153         return NULL;
```

```

154     len = strlen(str);
155     str_copy = strdup(str);
156     if (!str_copy)
157         return NULL;
158     if (len > 0 && str[len - 1] == '\n')
159         str_copy[len - 1] = '\0';
160     if (!str_copy)
161         return NULL;
162     result = process_tokens(str_copy, delim);
163     free(str_copy);
164     return result;
165 }

```

14.176.3.49 take_object()

```

void take_object (
    server_t * server,
    client_t * client,
    char ** buffer )

```

Definition at line 43 of file take_object.c.

```

44 {
45     resource_type_t type;
46
47     if (!server || !client || !client->player || arr_len(buffer) != 2)
48         return write_command_output(client->client_fd, "ko\n");
49     type = determine_type(buffer[1]);
50     if (type == COUNT)
51         return write_command_output(client->client_fd, "ko\n");
52     if (server->map[client->player->pos_y]
53         [client->player->pos_x].resources[type] > 0) {
54         update_resources(server, client, type);
55         add_item_to_inventory(client->player, type, 1);
56         command_pgt(server, client, type);
57         send_bct_to_all_graphical_clients(server, client->player->pos_x,
58             client->player->pos_y);
59         send_pin_to_all(server, client);
60         write_command_output(client->client_fd, "ok\n");
61     } else
62         write_command_output(client->client_fd, "ko\n");
63 }

```

14.176.3.50 tile_to_str()

```

char* tile_to_str (
    tile_t * tile )

```

Definition at line 81 of file tile_to_str.c.

```

82 {
83     char *player = repeat_word("player", tile->player_count);
84     char *resources = add_resources(tile->resources);
85     int p_len = strlen(player);
86     int r_len = strlen(resources);
87     char *res = malloc(p_len + r_len + 2);
88
89     if (!res)
90         server_err("Malloc failed res tile_to_str");
91     res[0] = '\0';
92     if (p_len)
93         strcat(res, player);
94     if (p_len && r_len)
95         strcat(res, " ");
96     if (r_len)
97         strcat(res, resources);
98     free(player);
99     free(resources);
100    return res;
101 }

```

14.176.3.51 write_command_output()

```

void write_command_output (
    int client_fd,
    char * msg )

```

Definition at line 75 of file parse_command_utils.c.

```

76 {
77     if (fcntl(client_fd, F_GETFD) == -1) {
78         perror("FD isn't up anymore\n");
79     } else {
80         write(client_fd, msg, strlen(msg));
81     }
82 }

```

14.177 /root/Desktop/Zappy/src/Server/include/egg.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [egg_s](#)

Typedefs

- typedef struct [server_s](#) [server_t](#)
- typedef struct [tile_s](#) [tile_t](#)
- typedef struct [egg_s](#) [egg_t](#)

Functions

- [egg_t * create_egg](#) (int egg_id, int pos_x, int pos_y, char *team)
- void [remove_egg](#) ([server_t](#) *server, int id, [tile_t](#) *tile)
- void [add_egg](#) ([server_t](#) *server, [egg_t](#) *egg)

14.177.1 Typedef Documentation

14.177.1.1 [egg_t](#)

typedef struct [egg_s](#) [egg_t](#)

14.177.1.2 [server_t](#)

typedef struct [server_s](#) [server_t](#)

Definition at line 1 of file egg.h.

14.177.1.3 [tile_t](#)

typedef struct [tile_s](#) [tile_t](#)

Definition at line 1 of file egg.h.

14.177.2 Function Documentation

14.177.2.1 add_egg()

```
void add_egg (
    server_t * server,
    egg_t * egg )
```

Definition at line 114 of file egg.c.

```
115 {
116     tile_t *tile = NULL;
117
118     if (server == NULL || egg == NULL)
119         return;
120     egg->next = server->eggs;
121     server->eggs = egg;
122     tile = &server->map[egg->pos_y][egg->pos_x];
123     add_egg_to_tile(tile, egg->egg_id);
124 }
```

14.177.2.2 create_egg()

```
egg_t* create_egg (
    int egg_id,
    int pos_x,
    int pos_y,
    char * team )
```

Definition at line 14 of file egg.c.

```
15 {
16     egg_t *new_egg = malloc(sizeof(egg_t));
17
18     if (new_egg == NULL)
19         return NULL;
20     new_egg->egg_id = egg_id;
21     new_egg->pos_x = pos_x;
22     new_egg->pos_y = pos_y;
23     new_egg->next = NULL;
24     new_egg->team_name = strdup(team);
25     if (new_egg->team_name == NULL)
26         server_err("Egg team allocation failed\n");
27     return new_egg;
28 }
```

14.177.2.3 remove_egg()

```
void remove_egg (
    server_t * server,
    int id,
    tile_t * tile )
```

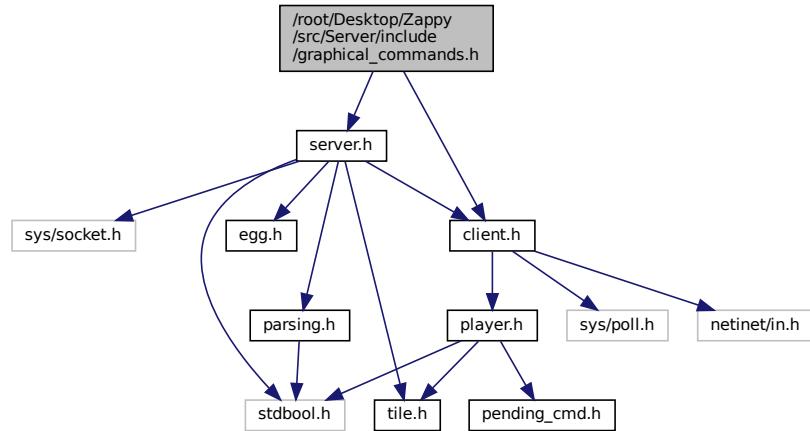
Definition at line 84 of file egg.c.

```
85 {
86     if (tile == NULL)
87         return;
88     remove_egg_from_tile(tile, id);
89     remove_egg_by_id(server, id);
90 }
```

14.178 /root/Desktop/Zappy/src/Server/include/graphical_commands.h File Reference

```
#include "server.h"
#include "client.h"
```

Include dependency graph for graphical_commands.h:



This graph shows which files directly or indirectly include this file:



Functions

- void [send_msz_command](#) (*server_t* *server, *client_t* *client)
- void [command_sgt](#) (*server_t* *server, *client_t* *client, char **buffer)
- void [send_tna_command](#) (*server_t* *server, *client_t* *client)
- void [send_pnw_command](#) (*server_t* *server, *client_t* *client, *client_t* *recipient)
- void [send_pin_command](#) (*server_t* *server, *client_t* *client, *client_t* *recipient)
- void [send_plv_command](#) (*server_t* *server, *client_t* *client, *client_t* *recipient)
- void [send_ebo_command](#) (*server_t* *server, int egg_id)
- void [command_pic](#) (*server_t* *server, int x, int y, int level)
- void [command_pbc](#) (*server_t* *server, *client_t* *client, char *buffer)
- void [command_seg](#) (*server_t* *server, const char *team_name)
- void [send_tile_content_to_one_client](#) (*server_t* *server, *client_t* *client)
- void [send_team_names_to_one_client](#) (*server_t* *server, *client_t* *client)
- void [send_all_player_info_to_one_client](#) (*server_t* *server, *client_t* *client)
- bool [send_ppo_command](#) (*server_t* *server, int id)
- void [send_plv_to_all](#) (*server_t* *server, *client_t* *client)
- void [send_pin_to_all](#) (*server_t* *server, *client_t* *client)
- void [send_edi_command](#) (*server_t* *server, int egg_id)
- void [send_smg_command](#) (*server_t* *server, const char *msg)

14.178.1 Function Documentation

14.178.1.1 command_pbc()

```
void command_pbc (
    server_t * server,
    client_t * client,
    char * buffer )
```

Definition at line 15 of file command_pbc.c.

```

16 {
17     graphical_client_t *graphical_client = NULL;
18     char *msg = NULL;
19     int size = 0;
20
21     if (!server || !client || !buffer ||
22         !server->graphical_clients || !client->player)
23         return;
24     size = sprintf(NULL, 0, "pbc %d %s", client->client_id, buffer);
25     msg = malloc(size + 1);
26     if (!msg)
27         return;
28     sprintf(msg, size + 1, "pbc %d %s", client->client_id, buffer);
29     graphical_client = server->graphical_clients;
30     while (graphical_client) {
31         write_command_output(graphical_client->client->client_fd, msg);
32         graphical_client = graphical_client->next;
33     }
34     free(msg);
35 }
```

14.178.1.2 command_pic()

```

void command_pic (
    server_t * server,
    int x,
    int y,
    int level )
```

Definition at line 36 of file command_pic.c.

```

37 {
38     graphical_client_t *current = NULL;
39     char *buffer = NULL;
40     tile_t *tile = NULL;
41
42     if (!server || !server->graphical_clients)
43         return;
44     if (y < 0 || x < 0 || y >= server->parsed_info->height
45         || x >= server->parsed_info->width)
46         return;
47     tile = &server->map[y][x];
48     buffer = get_pic_buffer(x, y, level, tile);
49     if (!buffer)
50         return;
51     current = server->graphical_clients;
52     while (current) {
53         write_command_output(current->client->client_fd, buffer);
54         current = current->next;
55     }
56     free(buffer);
57 }
```

14.178.1.3 command_seg()

```

void command_seg (
    server_t * server,
    const char * team_name )
```

Definition at line 15 of file command_seg.c.

```

16 {
17     graphical_client_t *current = NULL;
18     char *buffer = NULL;
19     int size = 0;
20
21     if (!server || !server->graphical_clients || !team_name)
22         return;
23     size = sprintf(NULL, 0, "seg %s\n", team_name);
24     buffer = malloc(size + 1);
25     if (!buffer)
26         return;
27     sprintf(buffer, size + 1, "seg %s\n", team_name);
28     current = server->graphical_clients;
29     while (current) {
30         write_command_output(current->client->client_fd, buffer);
31         current = current->next;
32     }
33     free(buffer);
34 }
```

14.178.1.4 command_sgt()

```
void command_sgt (
    server_t * server,
    client_t * client,
    char ** buffer )
```

Definition at line 16 of file command_sgt.c.

```
17 {
18     char *tmp = NULL;
19     int size = 0;
20
21     if (!server || !client || !server->graphical_clients ||
22         arr_len(buffer) != 1)
23         return write_command_output(client->client_fd, "spb\n");
24     tmp = NULL;
25     size = sprintf(NULL, 0, "sgt %d\n", server->parsed_info->frequence);
26     tmp = malloc(size + 1);
27     if (!tmp)
28         return;
29     sprintf(tmp, size + 1, "sgt %d\n", server->parsed_info->frequence);
30     write_command_output(client->client_fd, tmp);
31     free(tmp);
32 }
```

14.178.1.5 send_all_player_info_to_one_client()

```
void send_all_player_info_to_one_client (
    server_t * server,
    client_t * client )
```

Definition at line 64 of file command_pnw.c.

```
65 {
66     client_t *current = NULL;
67
68     if (!server || !client || !server->graphical_clients)
69         return;
70     current = server->client;
71     while (current) {
72         if (current->player && current->type == AI &&
73             current->is_fully_connected) {
74             send_pnw_command(server, current, client);
75             send_pin_command(server, current, client);
76             send_plv_command(server, current, client);
77         }
78         current = current->next;
79     }
80 }
```

14.178.1.6 send_ebo_command()

```
void send_ebo_command (
    server_t * server,
    int egg_id )
```

Definition at line 15 of file command_ebo.c.

```
16 {
17     graphical_client_t *current = NULL;
18     char *buffer;
19     int size;
20
21     if (!server || !server->graphical_clients)
22         return;
23     size = sprintf(NULL, 0, "ebo #%d\n", egg_id);
24     buffer = malloc(size + 1);
25     if (!buffer)
26         server_err("Buffer alloc in ebo command failed\n");
27     sprintf(buffer, "ebo #%d\n", egg_id);
28     current = server->graphical_clients;
29     while (current) {
30         write_command_output(current->client->client_fd, buffer);
31         current = current->next;
32     }
33     free(buffer);
34 }
```

14.178.1.7 send_edi_command()

```
void send_edi_command (
    server_t * server,
    int egg_id )
```

Definition at line 13 of file command_edi.c.

```
14 {
15     char *buffer = NULL;
16     int size = 0;
17     graphical_client_t *graphical_client = NULL;
18
19     if (!server || !server->graphical_clients)
20         return;
21     size = snprintf(NULL, 0, "edi #%d\n", egg_id);
22     buffer = malloc(size + 1);
23     if (!buffer)
24         return;
25     snprintf(buffer, size + 1, "edi #%d\n", egg_id);
26     graphical_client = server->graphical_clients;
27     while (graphical_client) {
28         write_command_output(graphical_client->client->client_fd, buffer);
29         graphical_client = graphical_client->next;
30     }
31     free(buffer);
32 }
```

14.178.1.8 send_msz_command()

```
void send_msz_command (
    server_t * server,
    client_t * client )
```

Definition at line 16 of file command_msz.c.

```
17 {
18     char *buffer = NULL;
19     int size = 0;
20
21     if (!server || !client)
22         return;
23     buffer = NULL;
24     size = snprintf(NULL, 0, "msz %d %d\n",
25                     server->parsed_info->width, server->parsed_info->height);
26     buffer = malloc(size + 1);
27     if (!buffer)
28         return;
29     snprintf(buffer, size + 1, "msz %d %d\n",
30             server->parsed_info->width, server->parsed_info->height);
31     write_command_output(client->client_fd, buffer);
32     free(buffer);
33 }
```

14.178.1.9 send_pin_command()

```
void send_pin_command (
    server_t * server,
    client_t * client,
    client_t * recipient )
```

Definition at line 55 of file command_pin.c.

```
56 {
57     char *buffer = get_buffer_pin_command(recipient);
58
59     (void)server;
60     if (!buffer || !server->graphical_clients)
61         return;
62     write_command_output(recipient->client_fd, buffer);
63     free(buffer);
64 }
```

14.178.1.10 send_pin_to_all()

```
void send_pin_to_all (
    server_t * server,
    client_t * client )
```

Definition at line 66 of file command_pin.c.

```
67 {
68     graphical_client_t *current = server->graphical_clients;
69
70     if (!server || !client || !server->graphical_clients)
71         return;
72     while (current) {
73         send_pin_command(server, client, current->client);
74         current = current->next;
75     }
76 }
```

14.178.1.11 send_plv_command()

```
void send_plv_command (
    server_t * server,
    client_t * client,
    client_t * recipient )
```

Definition at line 17 of file command_plv.c.

```
18 {
19     int size = 0;
20     char *buffer = NULL;
21
22     if (!server || !client || !client->player)
23         return;
24     size = snprintf(NULL, 0, "plv #%d %d\n",
25                     client->client_id,
26                     client->player->level);
27     buffer = malloc(size + 1);
28     if (!buffer)
29         return;
30     snprintf(buffer, size + 1, "plv #%d %d\n",
31             client->client_id,
32             client->player->level);
33     write_command_output(recipient->client_fd, buffer);
34     free(buffer);
35 }
```

14.178.1.12 send_plv_to_all()

```
void send_plv_to_all (
    server_t * server,
    client_t * client )
```

Definition at line 37 of file command_plv.c.

```
38 {
39     client_t *current = server->client;
40
41     if (!server || !client || !client->player)
42         return;
43     while (current) {
44         if (current->type == GRAPHICAL && current != client) {
45             send_plv_command(server, client, current);
46         }
47         current = current->next;
48     }
49 }
```

14.178.1.13 send_pnw_command()

```
void send_pnw_command (
    server_t * server,
    client_t * client,
    client_t * recipient )
```

Definition at line 28 of file command_pnw.c.

```
29 {
```

```

30     int size = 0;
31     char *buffer = NULL;
32
33     if (!server || !client || !client->player)
34         return;
35     size = calcul_size_pnw_command(client);
36     buffer = malloc(size + 1);
37     if (!buffer)
38         return;
39     snprintf(buffer, size + 1, "pnw %d %d %d %d %s\n",
40             client->client_id,
41             client->player->pos_x,
42             client->player->pos_y,
43             client->player->rotation,
44             client->player->level,
45             client->player->team_name);
46     write_command_output(recipient->client_fd, buffer);
47     free(buffer);
48 }

```

14.178.1.14 send_ppo_command()

```

bool send_ppo_command (
    server_t * server,
    int id )

```

Definition at line 48 of file command_ppo.c.

```

49 {
50     char *buffer = NULL;
51     client_t *tmp = NULL;
52
53     if (!server)
54         return false;
55     tmp = find_client_by_id(server, id);
56     if (!tmp || !tmp->player)
57         return false;
58     buffer = get_ppo_buffer(tmp);
59     for (client_t *cur = server->client; cur != NULL; cur = cur->next) {
60         if (cur->type == GRAPHICAL && cur->is_fully_connected) {
61             write_command_output(cur->client_fd, buffer);
62         }
63     }
64     free(buffer);
65     return true;
66 }

```

14.178.1.15 send_smg_command()

```

void send_smg_command (
    server_t * server,
    const char * msg )

```

Definition at line 16 of file command_smg.c.

```

17 {
18     graphical_client_t *current = NULL;
19     char *buffer = NULL;
20     int size = 0;
21
22     if (!server || !server->graphical_clients || !msg)
23         return;
24     size = sprintf(NULL, 0, "smg %s\n", msg);
25     buffer = malloc(size + 1);
26     if (!buffer)
27         return;
28     snprintf(buffer, size + 1, "smg %s\n", msg);
29     current = server->graphical_clients;
30     while (current) {
31         write_command_output(current->client->client_fd, buffer);
32         current = current->next;
33     }
34     free(buffer);
35 }

```

14.178.1.16 send_team_names_to_one_client()

```

void send_team_names_to_one_client (

```

```
    server_t * server,
    client_t * client )
```

14.178.1.17 send_tile_content_to_one_client()

```
void send_tile_content_to_one_client (
    server_t * server,
    client_t * client )
```

Definition at line 88 of file command_bct.c.

```
89 {
90     if (!server || !client || !server->graphical_clients)
91         return;
92     for (int y = 0; y < server->parsed_info->height; y++) {
93         for (int x = 0; x < server->parsed_info->width; x++) {
94             send_bct_command(server, client, x, y);
95         }
96     }
97 }
```

14.178.1.18 send_tna_command()

```
void send_tna_command (
    server_t * server,
    client_t * client )
```

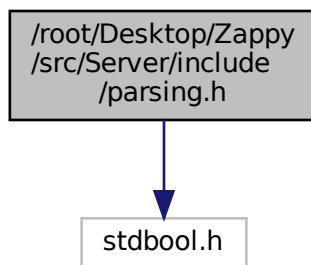
Definition at line 33 of file command_tna.c.

```
34 {
35     if (!server || !client)
36         return;
37     for (int i = 0; server->parsed_info->names[i]; i++) {
38         send_one_tna_command(server, client, server->parsed_info->names[i]);
39     }
40 }
```

14.179 /root/Desktop/Zappy/src/Server/include/parsing.h File Reference

```
#include <stdbool.h>
```

Include dependency graph for parsing.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [parsing_info_s](#)

Typedefs

- typedef struct [parsing_info_s](#) [parsing_info_t](#)

Functions

- void [parse_args](#) (int ac, char **av, [parsing_info_t](#) *parsed_info)
- void [parsing_error](#) (char *error_message, [parsing_info_t](#) *parsed_info)
- int [arr_len](#) (char **array)
- void [free_arr](#) (char **array)
- char ** [str_to_word_arr](#) (char *str, char *delim)

14.179.1 Typedef Documentation

14.179.1.1 [parsing_info_t](#)

```
typedef struct parsing\_info\_s parsing\_info\_t
```

14.179.2 Function Documentation

14.179.2.1 [arr_len\(\)](#)

```
int arr_len (
    char ** array )
Definition at line 13 of file array_function.c.
14 {
15     int i = 0;
16
17     if (!array)
18         return 0;
19     while (array[i])
20         i++;
21     return i;
22 }
```

14.179.2.2 [free_arr\(\)](#)

```
void free_arr (
    char ** array )
Definition at line 108 of file free.c.
109 {
110     if (!array)
111         return;
112     for (int i = 0; array[i] != NULL; i++) {
113         if (array[i] == NULL)
114             break;
115         free(array[i]);
116     }
117     free(array);
118 }
```

14.179.2.3 parse_args()

```
void parse_args (
    int ac,
    char ** av,
    parsing_info_t * parsed_info )
```

Definition at line 152 of file parsing.c.

```
153 {
154     parsed_info->port = 0;
155     parsed_info->width = 0;
156     parsed_info->height = 0;
157     parsed_info->client_nb = 0;
158     parsed_info->frequence = 100;
159     parsed_info->names = NULL;
160     for (int i = 1; i < ac; i++){
161         check_args(av, i, parsed_info);
162     }
163 }
```

14.179.2.4 parsing_error()

```
void parsing_error (
    char * error_message,
    parsing_info_t * parsed_info )
```

Definition at line 20 of file parsing_err.c.

```
21 {
22     if (parsed_info->names != NULL && parsed_info->names[0] != NULL)
23         free_names(parsed_info->names);
24     printf("%s\n", error_message);
25     display_help();
26     exit(84);
27 }
```

14.179.2.5 str_to_word_arr()

```
char** str_to_word_arr (
    char * str,
    char * delim )
```

Definition at line 146 of file array_function.c.

```
147 {
148     char *str_copy;
149     char **result;
150     int len = 0;
151
152     if (!str || !delim)
153         return NULL;
154     len = strlen(str);
155     str_copy = strdup(str);
156     if (!str_copy)
157         return NULL;
158     if (len > 0 && str[len - 1] == '\n')
159         str_copy[len - 1] = '\0';
160     if (!str_copy)
161         return NULL;
162     result = process_tokens(str_copy, delim);
163     free(str_copy);
164     return result;
165 }
```

14.180 /root/Desktop/Zappy/src/Server/include/pending_cmd.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [pending_cmd_s](#)

Typedefs

- typedef struct [server_s server_t](#)
- typedef struct [client_s client_t](#)
- typedef void(* [cmd_func_t](#)) ([server_t](#) *, [client_t](#) *, char **)
- typedef struct [pending_cmd_s pending_cmd_t](#)

14.180.1 Typedef Documentation

14.180.1.1 [client_t](#)

```
typedef struct client_s client_t
```

Definition at line 1 of file pending_cmd.h.

14.180.1.2 [cmd_func_t](#)

```
typedef void(* cmd_func_t) (server\_t *, client\_t *, char **)
```

Definition at line 15 of file pending_cmd.h.

14.180.1.3 [pending_cmd_t](#)

```
typedef struct pending_cmd_s pending_cmd_t
```

14.180.1.4 [server_t](#)

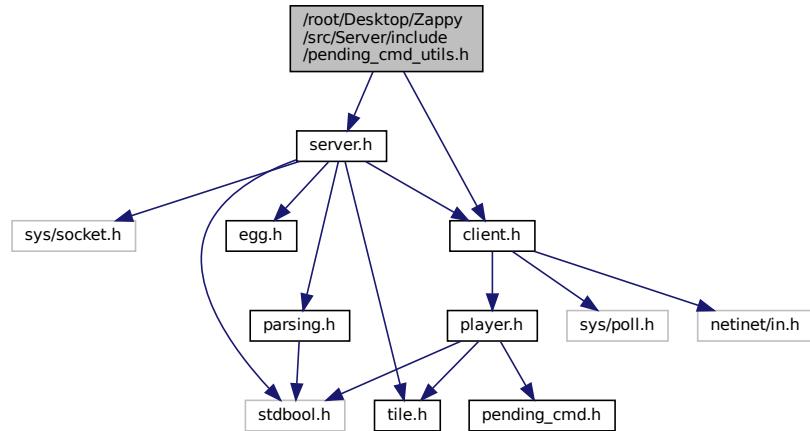
```
typedef struct server_s server_t
```

Definition at line 1 of file pending_cmd.h.

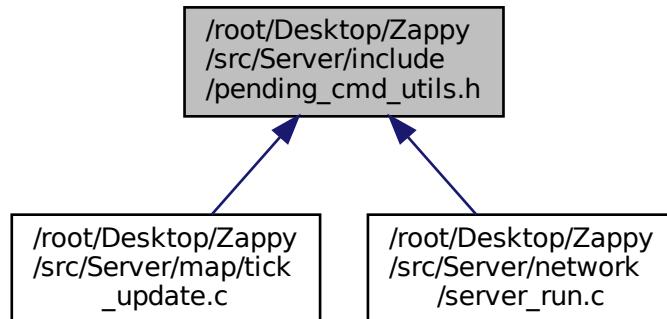
14.181 /root/Desktop/Zappy/src/Server/include/pending_cmd_utils.h File Reference

```
#include "server.h"  
#include "client.h"
```

Include dependency graph for pending_cmd_utils.h:



This graph shows which files directly or indirectly include this file:



Functions

- void [execute_pending_cmd](#) ([server_t](#) *server, [client_t](#) *client)

14.181.1 Function Documentation

14.181.1.1 execute_pending_cmd()

```
void execute_pending_cmd (
    server\_t * server,
    client\_t * client )
```

Definition at line 11 of file pending_cmd_utils.c.

```
12 {
13     pending\_cmd\_t *cmd;
14     char **args_array;
15 }
```

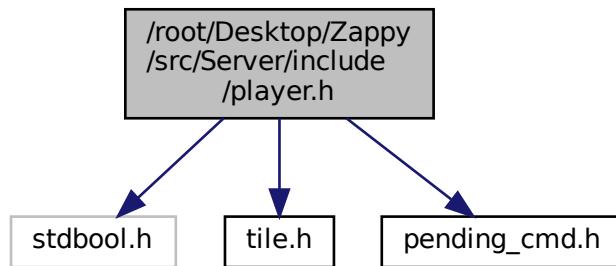
```

16     if (!client || !client->player || !client->player->pending_cmd)
17         return;
18     cmd = client->player->pending_cmd;
19     if (cmd->func && cmd->args) {
20         args_array = str_to_word_arr(cmd->args, " ");
21         cmd->func(server, client, args_array);
22         free_arr(args_array);
23     }
24     if (cmd->args)
25         free(cmd->args);
26     cmd->args = NULL;
27     cmd->func = NULL;
28 }
```

14.182 /root/Desktop/Zappy/src/Server/include/player.h File Reference

```
#include <stdbool.h>
#include "tile.h"
#include "pending_cmd.h"
```

Include dependency graph for player.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct `player_inventory_s`
- struct `player_s`

TypeDefs

- typedef struct `player_inventory_s` `player_inventory_t`
- typedef struct `player_s` `player_t`

Enumerations

- enum `rotation_e` { `UP` , `DOWN` , `LEFT` , `RIGHT` }

Functions

- void `init_player` (`player_t` *`player`, char *`playerTeam`)

- void `init_inventory (player_t *player)`
- void `free_inventory (player_t *player)`
- bool `add_item_to_inventory (player_t *player, resource_type_t type, int nb)`
- bool `remove_item_from_inventory (player_t *player, resource_type_t type, int nb)`
- bool `inventory_has_item (player_t *player, resource_type_t type)`
- int `how_many_in_inventory (player_t *player, resource_type_t type)`
- char * `get_inventory_content (player_t *player)`
- char * `get_resource_name (resource_type_t type)`

14.182.1 Typedef Documentation

14.182.1.1 player_inventory_t

```
typedef struct player_inventory_s player_inventory_t
```

14.182.1.2 player_t

```
typedef struct player_s player_t
```

14.182.2 Enumeration Type Documentation

14.182.2.1 rotation_e

```
enum rotation_e
```

Enumerator

UP	
DOWN	
LEFT	
RIGHT	

Definition at line 14 of file player.h.

```
14
15     UP,
16     DOWN,
17     LEFT,
18     RIGHT
19 };
```

14.182.3 Function Documentation

14.182.3.1 add_item_to_inventory()

```
bool add_item_to_inventory (
    player_t * player,
    resource_type_t type,
    int nb )
```

Definition at line 33 of file player_inventory.c.

```
34 {
35     if (!player || q < 0 || type >= COUNT)
36         return false;
37     player->inventory[type] += q;
38     return true;
39 }
```

14.182.3.2 free_inventory()

```
void free_inventory (
    player_t * player )
```

Definition at line 23 of file player_inventory.c.

```
24 {
25     if (!player)
26         return;
27     for (int i = 0; i < COUNT; i++) {
28         player->inventory[i] = 0;
29     }
30     player->inventory_size = 0;
31 }
```

14.182.3.3 get_inventory_content()

```
char* get_inventory_content (
    player_t * player )
```

Definition at line 67 of file player_inventory_display.c.

```
68 {
69     char *content = strdup("[");
70     size_t content_length = 1;
71     int quantity;
72
73     if (!player)
74         return strdup("[");
75     for (int i = 0; i < COUNT; i++) {
76         quantity = 0;
77         if (player->inventory[i])
78             quantity = player->inventory[i];
79         process_one_item(i, quantity, &content, &content_length);
80         if (!content)
81             return NULL;
82         add_separator(&content, i == COUNT - 1);
83     }
84     return content;
85 }
```

14.182.3.4 get_resource_name()

```
char* get_resource_name (
    resource_type_t type )
```

Definition at line 12 of file player_inventory_utils.c.

```
13 {
14     switch (type) {
15         case FOOD:
16             return strdup("food");
17         case LINEMATE:
18             return strdup("linemate");
19         case DERAUMERE:
20             return strdup("deraumere");
21         case SIBUR:
22             return strdup("sibur");
23         case MENDIANE:
24             return strdup("mendiane");
25         case PHIRAS:
26             return strdup("phiras");
27         case THYSTAME:
28             return strdup("thystame");
29         default:
30             return NULL;
31     }
32 }
```

14.182.3.5 how_many_in_inventory()

```
int how_many_in_inventory (
    player_t * player,
    resource_type_t type )
```

Definition at line 41 of file player_inventory_utils.c.

```
42 {
43     if (!player || type >= COUNT)
44         return 0;
45     if (!player->inventory[type])
46         return 0;
47     return player->inventory[type];
48 }
```

14.182.3.6 init_inventory()

```
void init_inventory (
    player_t * player )
```

Definition at line 13 of file player_inventory.c.

```
14 {
15     if (!player)
16         return;
17     for (int i = 0; i < COUNT; i++) {
18         player->inventory[i] = 0;
19     }
20     player->inventory[FOOD] = 10;
21 }
```

14.182.3.7 init_player()

```
void init_player (
    player_t * player,
    char * playerTeam )
```

Definition at line 73 of file player_init.c.

```
74 {
75     player->pos_x = 0;
76     player->pos_y = 0;
77     player->rotation = UP;
78     player->busy_until = 0;
79     player->queue_size = 0;
80     player->command_queue = calloc(10, sizeof(char *));
81     if (player->command_queue == NULL)
82         server_err("Malloc failed for command queue");
83     player->level = 1;
84     player->life = 126;
85     if (player_team != NULL) {
86         init_player_team(player, player_team);
87     } else
88         player->team_name = NULL;
89     init_inventory(player);
90     player->is_in_incantation = false;
91     player->incantation_leader_id = -1;
92     init_pending(player);
93 }
```

14.182.3.8 inventory_has_item()

```
bool inventory_has_item (
    player_t * player,
    resource_type_t type )
```

Definition at line 34 of file player_inventory_utils.c.

```
35 {
36     if (!player || type >= COUNT)
37         return false;
38     return player->inventory[type] > 0;
39 }
```

14.182.3.9 remove_item_from_inventory()

```
bool remove_item_from_inventory (
    player_t * player,
    resource_type_t t,
    int nb )
```

Definition at line 41 of file player_inventory.c.

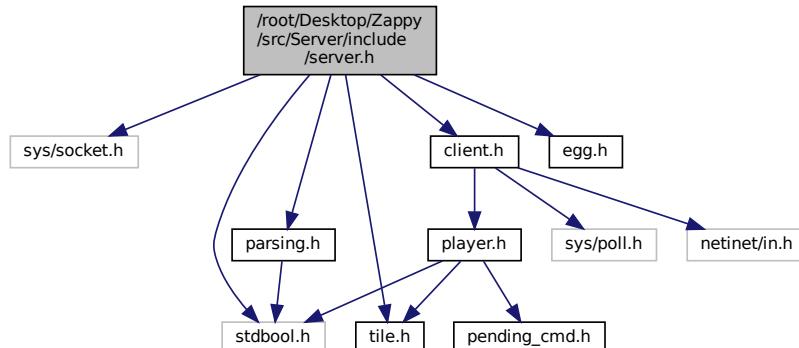
```

42 {
43     if (!player || nb <= 0 || type >= COUNT)
44         return false;
45     if (player->inventory[type] == 0) {
46         printf("Item not found in inventory.\n");
47         return false;
48     }
49     if (player->inventory[type] < nb)
50         printf("Warning: Not enough quantity to remove, remove all\n");
51     player->inventory[type] -= nb;
52     if (player->inventory[type] < 0)
53         player->inventory[type] = 0;
54     return true;
55 }
```

14.183 /root/Desktop/Zappy/src/Server/include/server.h File Reference

```
#include <sys/socket.h>
#include <stdbool.h>
#include "client.h"
#include "parsing.h"
#include "tile.h"
#include "egg.h"
```

Include dependency graph for server.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [poll_manager_s](#)
Structure de gestion du polling non-bloquant.
- struct [server_s](#)
Structure principale du serveur Zappy.

Typedefs

- typedef struct [poll_manager_s](#) [poll_manager_t](#)
Structure de gestion du polling non-bloquant.
- typedef struct [server_s](#) [server_t](#)
Structure principale du serveur Zappy.

Functions

- void `update_game_tick (server_t *server)`
Met à jour le tick de jeu et traite les actions en attente.
- void `finish_incantation (server_t *server, client_t *client)`
Finalise une incantation en cours.
- void `display_help (void)`
Affiche l'aide d'utilisation du serveur.
- void `remove_fd (server_t *server, int fd)`
Supprime un descripteur de fichier du gestionnaire de polling.
- void `add_fd (server_t *server, int fd)`
Ajoute un descripteur de fichier au gestionnaire de polling.
- void `create_server (server_t *server, parsing_info_t *parsed_info)`
Initialise et configure le serveur.
- void `server_err (char *msg)`
Affiche un message d'erreur et termine le programme.
- void `print_co (char *client_ip, struct sockaddr_in *client_addr, client_t *new_client)`
Affiche les informations de connexion d'un nouveau client.
- void `check_client (server_t *server)`
Vérifie et traite les nouvelles connexions clients.
- void `create_map (server_t *server, parsing_info_t *parsed_info)`
Initialise et génère la carte de jeu.
- void `init_new_player_pos (server_t *server, client_t *new_client)`
Initialise la position d'un nouveau joueur sur la carte.
- void `process_next_queued_command (server_t *server, client_t *client)`
Traite la prochaine commande en file d'attente pour un client.
- void `add_to_command_queue (server_t *server, client_t *client, char *command)`
Ajoute une commande à la file d'attente d'un client.
- void `free_node (client_t *node, server_t *server)`
Libère la mémoire d'un nud client et le supprime du serveur.
- void `init_server_eggs (server_t *n_server)`
Initialise les ufs de départ sur le serveur.
- void `free_all (server_t *server, parsing_info_t *parsed_info)`
Libère toute la mémoire allouée par le serveur.
- int `count_team (server_t *n_server)`
Compte le nombre d'équipes configurées.

14.183.1 Typedef Documentation

14.183.1.1 poll_manager_t

```
typedef struct poll_manager_s poll_manager_t
```

Structure de gestion du polling non-bloquant.

Cette structure gère les descripteurs de fichiers pour le polling et optimise les performances en évitant les reconstructions inutiles.

14.183.1.2 server_t

```
typedef struct server_s server_t
```

Structure principale du serveur Zappy.

Cette structure contient tous les éléments nécessaires au fonctionnement du serveur : connexions clients, état du jeu, configuration, etc.

14.183.2 Function Documentation

14.183.2.1 add_fd()

```
void add_fd (
    server_t * server,
    int fd )
```

Ajoute un descripteur de fichier au gestionnaire de polling.

Parameters

<i>server</i>	Pointeur vers la structure du serveur
<i>fd</i>	Descripteur de fichier à ajouter

Definition at line 95 of file connection.c.

```
96 {
97     static int next_id = 0;
98     client_t *new_c = init_new_client(fd);
99     client_t *current;
100
101    if (server->client == NULL) {
102        server->client = new_c;
103        new_c->client_id = -1;
104        return;
105    }
106    current = server->client;
107    while (current->next != NULL) {
108        current = current->next;
109    }
110    new_c->client_id = next_id;
111    next_id++;
112    current->next = new_c;
113 }
```

14.183.2.2 add_to_command_queue()

```
void add_to_command_queue (
    server_t * server,
    client_t * client,
    char * command )
```

Ajoute une commande à la file d'attente d'un client.

Parameters

<i>server</i>	Pointeur vers la structure du serveur
<i>client</i>	Client destinataire de la commande
<i>command</i>	Commande à ajouter à la file

Definition at line 14 of file queue_connection.c.

```
15 {
16     if (!client || !client->player || !command ||
17         client->player->queue_size >= 10 || !client->player->command_queue)
18         return;
19     client->player->command_queue[client->player->queue_size] =
20         strdup(command);
21     client->player->queue_size++;
22     if (client->player->queue_size >= 10 && server && server->poll_manager)
23         server->poll_manager->needs_rebuild = true;
24 }
```

14.183.2.3 check_client()

```
void check_client (
    server_t * server )
```

Vérifie et traite les nouvelles connexions clients.

Parameters

<i>server</i>	Pointeur vers la structure du serveur
---------------	---------------------------------------

Definition at line 151 of file server_run.c.

```
152 {
153     int size = server->nfds + 1;
154
155     setup_poll_manager(server->poll_manager, size);
156     handle_game_tick(server);
157     if (server->poll_manager->needs_rebuild)
158         fill_poll_array(server, server->poll_manager);
159     if (poll(server->poll_manager->fds, size, 10) > 0)
160         poll_client(server, server->poll_manager);
161 }
```

14.183.2.4 count_team()

```
int count_team (
    server_t * n_server )
```

Compte le nombre d'équipes configurées.

Parameters

<i>n_server</i>	Pointeur vers la structure du serveur
-----------------	---------------------------------------

Returns

Nombre d'équipes

Definition at line 19 of file connection_utils.c.

```
20 {
21     int i = 0;
22
23     for ( ; n_server->parsed_info->names[i]; i++)
24     }
25     return i;
26 }
```

14.183.2.5 create_map()

```
void create_map (
    server_t * server,
    parsing_info_t * parsed_info )
```

Initialise et génère la carte de jeu.

Parameters

<i>server</i>	Pointeur vers la structure du serveur
<i>parsed_info</i>	Informations de configuration pour la carte

Definition at line 13 of file map_creation.c.

```
14 {
15     server->map = malloc(sizeof(tile_t *) * parsed_info->height);
16     if (server->map == NULL)
17         exit(84);
18     for (int y = 0; y < parsed_info->height; y++) {
19         server->map[y] = malloc(sizeof(tile_t) * parsed_info->width);
20         if (server->map[y] == NULL)
21             exit(84);
22         for (int x = 0; x < parsed_info->width; x++) {
23             tile_init(&server->map[y][x]);
24         }
25     }
```

```

26     distribute_resources(server->map, server,
27         server->total_resources, server->current_resources);
28 }
```

14.183.2.6 create_server()

```

void create_server (
    server_t * server,
    parsing_info_t * parsed_info )
```

Initialise et configure le serveur.

Parameters

<i>server</i>	Pointeur vers la structure du serveur à initialiser
<i>parsed_info</i>	Informations de configuration parsées

Definition at line 189 of file connection.c.

```

190 {
191     init_server(server, parsed_info);
192     init_server_socket(server, parsed_info);
193 }
```

14.183.2.7 display_help()

```

void display_help (
    void )
```

Affiche l'aide d'utilisation du serveur.

Affiche l'aide d'utilisation du serveur.

Definition at line 13 of file help.c.

```

14 {
15     printf("USAGE: ./zappy_server -p port -x width -y height -n name1 "
16         "'name2 ... -c clientsNb -f freq --auto-start on|off --display-eggs "
17         "true|false\n");
18     printf("[ -v | --verbose]--game_duration time\n");
19     printf("\tport\tis the port number\n");
20     printf("\twidht\tis the width of the world\n");
21     printf("\theight\tis the height of the world\n");
22     printf("\tnameX\tis the name of the team X\n");
23     printf("\tclientsNb\tis the number of authorized clients per team\n");
24     printf("\tfreq\tis the reciprocal of time unit for execution of "
25         "actions\n");
26     printf("\tauto-start\tdoes the greeting is send automaticly\n");
27     printf("\tdisplay-eggs\teggs are visible and destructible\n");
28     printf("\tmatch_duration\tis the duration of the match in seconds\n");
29 }
```

14.183.2.8 finish_incantation()

```

void finish_incantation (
    server_t * server,
    client_t * client )
```

Finalise une incantation en cours.

Parameters

<i>server</i>	Pointeur vers la structure du serveur
<i>client</i>	Client effectuant l'incantation

Definition at line 81 of file finish_incantation.c.

```

82 {
83     tile_t *tile = &server->map[client->player->pos_y][client->player->pos_x];
84     int old_level;
85
86     if (!client || !client->player || !client->player->is_in_incantation ||
87         client->player->busy_until > server->current_tick)
```

```

88     return;
89     old_level = client->player->level;
90     client->player->is_in_incantation = false;
91     if (!can_start_incantation(server, client)) {
92         command_pie(server, client->player->pos_x,
93                     client->player->pos_y, 0);
94         return handle_incantation_failure(client);
95     }
96     handle_incantation_success(client, tile, old_level, server);
97     command_pie(server, client->player->pos_x,
98                 client->player->pos_y, 1);
99     send_plv_to_all(server, client);
100 }
```

14.183.2.9 free_all()

```
void free_all (
    server_t * server,
    parsing_info_t * parsed_info )
```

Libère toute la mémoire allouée par le serveur.

Parameters

<i>server</i>	Pointeur vers la structure du serveur
<i>parsed_info</i>	Informations de configuration à libérer

Definition at line 90 of file free.c.

```

91 {
92     if (!server)
93         return;
94     free_client(server);
95     free_eggs(server);
96     free_map(server, parsed_info);
97     if (server->poll_manager)
98         free_poll_mana(server);
99     if (parsed_info)
100         free_parsed_info(parsed_info);
101     if (server->parsed_info) {
102         free_parsed_info(server->parsed_info);
103         free(server->parsed_info);
104         server->parsed_info = NULL;
105     }
106 }
```

14.183.2.10 free_node()

```
void free_node (
    client_t * node,
    server_t * server )
```

Libère la mémoire d'un nud client et le supprime du serveur.

Parameters

<i>node</i>	Nud client à supprimer
<i>server</i>	Pointeur vers la structure du serveur

Definition at line 32 of file client_node.c.

```

33 {
34     if (node->type == GRAPHICAL)
35         remove_graphic_client(server, node);
36     if (node->client_poll)
37         free(node->client_poll);
38     if (node->client_add)
39         free(node->client_add);
40     if (node->player) {
41         cleanup_player_client(node);
42         cleanup_pending(node->player);
43         free(node->player);
44     }
45     free(node);
```

```
46 }
```

14.183.2.11 init_new_player_pos()

```
void init_new_player_pos (
    server_t * server,
    client_t * new_client )
```

Initialise la position d'un nouveau joueur sur la carte.

Parameters

<i>server</i>	Pointeur vers la structure du serveur
<i>new_client</i>	Client dont le joueur doit être positionné

Initialise la position d'un nouveau joueur sur la carte.

Definition at line 33 of file player_init.c.

```
34 {
35     int random_x = rand() % server->parsed_info->width;
36     int random_y = rand() % server->parsed_info->height;
37     egg_t *team_egg;
38
39     if (new_client->player->team_name != NULL) {
40         team_egg = find_egg_for_team(server, new_client->player->team_name);
41         if (team_egg != NULL) {
42             new_client->player->pos_x = team_egg->pos_x;
43             new_client->player->pos_y = team_egg->pos_y;
44             tile_add_player(&server->map[team_egg->pos_y][team_egg->pos_x],
45                             new_client->client_id);
46             send_ebo_command(server, team_egg->egg_id);
47             remove_egg(server, team_egg->egg_id,
48                         &server->map[team_egg->pos_y][team_egg->pos_x]);
49             return;
50         }
51     }
52     new_client->player->pos_x = random_x;
53     new_client->player->pos_y = random_y;
54     tile_add_player(&server->map[random_y][random_x], new_client->client_id);
55 }
```

14.183.2.12 init_server_eggs()

```
void init_server_eggs (
    server_t * n_server )
```

Initialise les ufs de départ sur le serveur.

Parameters

<i>n_server</i>	Pointeur vers la structure du serveur
-----------------	---------------------------------------

Definition at line 40 of file connection_utils.c.

```
41 {
42     int team_count = count_team(n_server);
43     int eggs_per_team = n_server->parsed_info->client_nb;
44     int egg_id = 0;
45
46     for (int i = 0; i < team_count; i++) {
47         for (int j = 0; j < eggs_per_team; j++) {
48             create_server_egg(n_server, egg_id, i);
49             egg_id++;
50         }
51     }
52 }
```

14.183.2.13 print_co()

```
void print_co (
    char * client_ip,
```

```
    struct sockaddr_in * client_addr,
    client_t * new_client )
```

Affiche les informations de connexion d'un nouveau client.

Parameters

<i>client_ip</i>	Adresse IP du client
<i>client_addr</i>	Structure d'adresse du client
<i>new_client</i>	Structure du nouveau client

Definition at line 23 of file connection_err.c.

```
25 {
26     inet_ntop(AF_INET, &(client_addr->sin_addr), client_ip, INET_ADDRSTRLEN);
27     printf("New client connected from %s:%d with ID %d.\n",
28            client_ip, ntohs(client_addr->sin_port), new_client->client_id);
29 }
```

14.183.2.14 process_next_queued_command()

```
void process_next_queued_command (
    server_t * server,
    client_t * client )
```

Traite la prochaine commande en file d'attente pour un client.

Parameters

<i>server</i>	Pointeur vers la structure du serveur
<i>client</i>	Client dont la commande doit être traitée

Definition at line 33 of file queue_connection.c.

```
34 {
35     char *next_command;
36     bool was_full;
37
38     if (!client || !client->player || !server ||
39         client->player->queue_size <= 0 || !client->player->command_queue)
40         return;
41     was_full = (client->player->queue_size >= 10);
42     next_command = client->player->command_queue[0];
43     for (int i = 0; i < client->player->queue_size - 1; i++) {
44         client->player->command_queue[i] =
45             client->player->command_queue[i + 1];
46     }
47     client->player->command_queue[client->player->queue_size - 1] = NULL;
48     client->player->queue_size--;
49     check_rebuild(was_full, client, server);
50     if (next_command) {
51         execute_com(server, client, next_command);
52         free(next_command);
53         next_command = NULL;
54     }
55 }
```

14.183.2.15 remove_fd()

```
void remove_fd (
    server_t * server,
    int fd )
```

Supprime un descripteur de fichier du gestionnaire de polling.

Parameters

<i>server</i>	Pointeur vers la structure du serveur
<i>fd</i>	Descripteur de fichier à supprimer

Definition at line 61 of file connection.c.

```
62 {
63     if (server->client == NULL)
64         return;
65     if (remove_head_client(server, fd))
66         return;
67     remove_other_client(server, fd);
68     close(fd);
69 }
```

14.183.2.16 server_err()

```
void server_err (
    char * msg )
```

Affiche un message d'erreur et termine le programme.

Parameters

<i>msg</i>	Message d'erreur à afficher
------------	-----------------------------

Definition at line 17 of file connection_err.c.

```
18 {
19     perror(msg);
20     exit(84);
21 }
```

14.183.2.17 update_game_tick()

```
void update_game_tick (
    server_t * server )
```

Met à jour le tick de jeu et traite les actions en attente.

Parameters

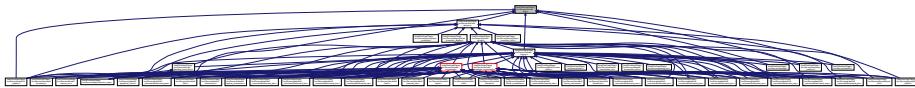
<i>server</i>	Pointeur vers la structure du serveur
---------------	---------------------------------------

Definition at line 29 of file tick_update.c.

```
30 {
31     client_t *current = server->client;
32     client_t *next;
33
34     server->current_tick++;
35     if (server->current_tick % 20 == 0)
36         respawn_resources(server->map, server, server->total_resources,
37                             server->current_resources);
38     if (current != NULL)
39         current = current->next;
40     while (current != NULL) {
41         next = current->next;
42         if (current->player && current->player->pending_cmd &&
43             current->player->busy_until <= server->current_tick)
44             execute_pending_cmd(server, current);
45         if (tick_check(server, current) == false && current->player &&
46             current->player->busy_until <= server->current_tick &&
47             current->player->queue_size > 0)
48             process_next_queued_command(server, current);
49         current = next;
50     }
51 }
```

14.184 /root/Desktop/Zappy/src/Server/include/tile.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [tile_s](#)
- struct [resource_dist_s](#)

Macros

- #define [FOOD_D](#) 0.5
- #define [LINEMATE_D](#) 0.3
- #define [DERAUMERE_D](#) 0.15
- #define [SIBUR_D](#) 0.1
- #define [MENDIANE_D](#) 0.1
- #define [PHIRAS_D](#) 0.08
- #define [THYSTAME_D](#) 0.05

Typedefs

- typedef struct [server_s](#) [server_t](#)
- typedef enum [resource_type](#) [resource_type_t](#)
- typedef struct [tile_s](#) [tile_t](#)
- typedef struct [resource_dist_s](#) [resource_dist_t](#)

Enumerations

- enum [resource_type](#) {
 [FOOD](#) = 0 , [LINEMATE](#) , [DERAUMERE](#) , [SIBUR](#) ,
 [MENDIANE](#) , [PHIRAS](#) , [THYSTAME](#) , COUNT }

Functions

- void [tile_init](#) ([tile_t](#) *[tile](#))
- void [tile_free](#) ([tile_t](#) *[tile](#))
- int [tile_add_player](#) ([tile_t](#) *[tile](#), int [player_id](#))
- int [tile_remove_player](#) ([tile_t](#) *[tile](#), int [player_id](#))
- void [add_egg_to_tile](#) ([tile_t](#) *[tile](#), int [egg_id](#))
- void [remove_egg_from_tile](#) ([tile_t](#) *[tile](#), int [egg_id](#))
- void [distribute_resources](#) ([tile_t](#) **[map](#), [server_t](#) *[server](#), int *[total_resources](#), int *[current_resources](#))
- void [respawn_resources](#) ([tile_t](#) **[map](#), [server_t](#) *[server](#), int *[total_resources](#), int *[current_resources](#))

14.184.1 Macro Definition Documentation

14.184.1.1 DERAUMERE_D

```
#define DERAUMERE_D 0.15
Definition at line 12 of file tile.h.
```

14.184.1.2 FOOD_D

```
#define FOOD_D 0.5
```

Definition at line 10 of file tile.h.

14.184.1.3 LINEMATE_D

```
#define LINEMATE_D 0.3
```

Definition at line 11 of file tile.h.

14.184.1.4 MENDIANE_D

```
#define MENDIANE_D 0.1
```

Definition at line 14 of file tile.h.

14.184.1.5 PHIRAS_D

```
#define PHIRAS_D 0.08
```

Definition at line 15 of file tile.h.

14.184.1.6 SIBUR_D

```
#define SIBUR_D 0.1
```

Definition at line 13 of file tile.h.

14.184.1.7 THYSTAME_D

```
#define THYSTAME_D 0.05
```

Definition at line 16 of file tile.h.

14.184.2 Typedef Documentation

14.184.2.1 resource_dist_t

```
typedef struct resource_dist_s resource_dist_t
```

14.184.2.2 resource_type_t

```
typedef enum resource_type_s resource_type_t
```

14.184.2.3 server_t

```
typedef struct server_s server_t
```

Definition at line 1 of file tile.h.

14.184.2.4 tile_t

```
typedef struct tile_s tile_t
```

14.184.3 Enumeration Type Documentation

14.184.3.1 resource_type

```
enum resource_type
```

Enumerator

FOOD	
LINEMATE	
DERAUMERE	
SIBUR	
MENDIANE	
PHIRAS	
THYSTAME	
COUNT	

Definition at line 19 of file tile.h.

```
19      {
20      FOOD = 0,
21      LINEMATE,
22      DERAUMERE,
23      SIBUR,
24      MENDIANE,
25      PHIRAS,
26      THYSTAME,
27      COUNT
28 } resource_type_t;
```

14.184.4 Function Documentation

14.184.4.1 add_egg_to_tile()

```
void add_egg_to_tile (
    tile_t * tile,
    int egg_id )
```

Definition at line 92 of file egg.c.

```
93 {
94     int new_capacity;
95     int *new_egg_ids;
96
97     if (tile == NULL)
98         return;
99     if (tile->egg_count >= tile->egg_capacity) {
100         if (tile->egg_capacity == 0)
101             new_capacity = 4;
102         else
103             new_capacity = tile->egg_capacity * 2;
104         new_egg_ids = realloc(tile->egg_ids, sizeof(int) * new_capacity);
105         if (new_egg_ids == NULL)
106             return;
107         tile->egg_ids = new_egg_ids;
108         tile->egg_capacity = new_capacity;
109     }
110     tile->egg_ids[tile->egg_count] = egg_id;
111     tile->egg_count++;
112 }
```

14.184.4.2 distribute_resources()

```
void distribute_resources (
    tile_t ** map,
    server_t * server,
```

```

        int * total_resources,
        int * current_resources )
Definition at line 61 of file resources_map.c.
63 {
64     int total;
65     resource_dist_t dist;
66     static double resource_densities[COUNT] = {FOOD_D, LINEMATE_D, DERAUMERE_D,
67         SIBUR_D, MENDIANE_D, PHIRAS_D, THYSTAME_D};
68
69     init_resource_dist(server->parsed_info->width,
70         server->parsed_info->height, map, &dist);
71     for (int res = 0; res < COUNT; ++res) {
72         total = (int)(server->parsed_info->width *
73             server->parsed_info->height * resource_densities[res] + 0.5);
74         if (total < 1)
75             total = 1;
76         total_resources[res] = total;
77         current_resources[res] = total;
78         distribute_one_resource(&dist, res, total, server);
79     }
80     free(dist.tile_indices);
81 }
```

14.184.4.3 remove_egg_from_tile()

```
void remove_egg_from_tile (
    tile_t * tile,
    int egg_id )
```

Definition at line 77 of file egg.c.

```
78 {
79     if (tile == NULL)
80         return;
81     remove_id_from_array(&tile->egg_ids, &tile->egg_count, egg_id);
82 }
```

14.184.4.4 respawn_resources()

```
void respawn_resources (
    tile_t ** map,
    server_t * server,
    int * total_resources,
    int * current_resources )
```

Definition at line 83 of file resources_map.c.

```
85 {
86     resource_dist_t dist;
87     int missing;
88
89     init_resource_dist(server->parsed_info->width,
90         server->parsed_info->height, map, &dist);
91     for (int res = 0; res < COUNT; ++res) {
92         missing = total_resources[res] - current_resources[res];
93         if (missing > 0) {
94             distribute_one_resource(&dist, res, missing, server);
95             current_resources[res] += missing;
96         }
97     }
98     send_smg_command(server, "Resources respawned");
99     free(dist.tile_indices);
100 }
```

14.184.4.5 tile_add_player()

```
int tile_add_player (
    tile_t * tile,
    int player_id )
```

Definition at line 37 of file tile.c.

```
38 {
39     int new_capacity;
40     int *new_ids;
41
42     if (tile->player_count >= tile->player_capacity) {
```

```

43     if (tile->player_capacity == 0)
44         new_capacity = 4;
45     else
46         new_capacity = tile->player_capacity * 2;
47     new_ids = realloc(tile->player_ids, new_capacity * sizeof(int));
48     if (!new_ids)
49         return -1;
50     tile->player_ids = new_ids;
51     tile->player_capacity = new_capacity;
52 }
53 tile->player_ids[tile->player_count] = player_id;
54 tile->player_count++;
55 return 0;
56 }
```

14.184.4.6 tile_free()

```
void tile_free (
    tile_t * tile )
```

Definition at line 25 of file tile.c.

```

26 {
27     free(tile->player_ids);
28     tile->player_ids = NULL;
29     tile->player_count = 0;
30     tile->player_capacity = 0;
31     free(tile->egg_ids);
32     tile->egg_ids = NULL;
33     tile->egg_count = 0;
34     tile->egg_capacity = 0;
35 }
```

14.184.4.7 tile_init()

```
void tile_init (
    tile_t * tile )
```

Definition at line 14 of file tile.c.

```

15 {
16     tile->player_ids = NULL;
17     tile->player_count = 0;
18     tile->player_capacity = 0;
19     tile->egg_ids = NULL;
20     tile->egg_count = 0;
21     tile->egg_capacity = 0;
22     memset(tile->resources, 0, sizeof(tile->resources));
23 }
```

14.184.4.8 tile_remove_player()

```
int tile_remove_player (
    tile_t * tile,
    int player_id )
```

Definition at line 58 of file tile.c.

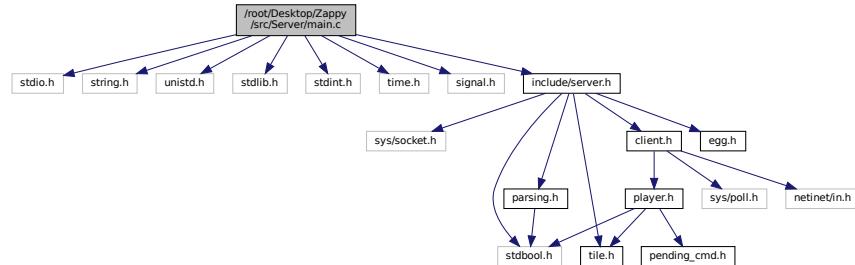
```

59 {
60     for (int i = 0; i < tile->player_count; ++i) {
61         if (tile->player_ids[i] == player_id) {
62             tile->player_ids[i] = tile->player_ids[tile->player_count - 1];
63             tile->player_count--;
64             return 0;
65         }
66     }
67     return -1;
68 }
```

14.185 /root/Desktop/Zappy/src/Server/main.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <stdlib.h>
```

```
#include <stdint.h>
#include <time.h>
#include <signal.h>
#include "include/server.h"
Include dependency graph for main.c:
```



Functions

- static int * [get_running_flag](#) (void)
- static void [sigint_handler](#) (int signum)
- static void [sigpipe_handler](#) (int signum)
- static int [setup_signal_handler](#) (void)
- static int [should_continue_running](#) (void)
- static int [check_mini_args](#) (int ac, char **av)
- static void [disp_args](#) (parsing_info_t *parsed_info)
- static void [server_loop](#) (server_t *server)
- static void [set_rdm_seed](#) (void)
- int [main](#) (int ac, char **av)

Main function for the zappy server.

14.185.1 Function Documentation

14.185.1.1 [check_mini_args\(\)](#)

```
static int check_mini_args (
    int ac,
    char ** av ) [static]
Definition at line 68 of file main.c.
69 {
70     int count = 0;
71
72     for (int i = 0; i < ac; i++) {
73         if (strcmp(av[i], "-p") == 0 || strcmp(av[i], "-x") == 0 ||
74             strcmp(av[i], "-y") == 0 || strcmp(av[i], "-n") == 0 ||
75             strcmp(av[i], "-c") == 0)
76             count++;
77     }
78     if (count == 5)
79         return 0;
80     return 1;
81 }
```

14.185.1.2 [disp_args\(\)](#)

```
static void disp_args (
    parsing_info_t * parsed_info ) [static]
```

Definition at line 83 of file main.c.

```

84 {
85     printf("port = %d\n", parsed_info->port);
86     printf("width = %d\n", parsed_info->width);
87     printf("height = %d\n", parsed_info->height);
88     printf("clients_nb = %d\n", parsed_info->client_nb);
89     printf("freq = %d\n", parsed_info->frequence);
90     printf("name :");
91     for (int i = 0; parsed_info->names[i] != NULL; i++) {
92         printf(" %s", parsed_info->names[i]);
93     }
94     printf("\n");
95 }
```

14.185.1.3 get_running_flag()

```
static int* get_running_flag (
    void ) [static]
```

Definition at line 16 of file main.c.

```

17 {
18     static int keep_running = 1;
19
20     return &keep_running;
21 }
```

14.185.1.4 main()

```
int main (
    int ac,
    char ** av )
```

Main function for the zappy server.

Parameters

<i>argc</i>	Number of arguments
<i>argv</i>	Array of arguments

Returns

0 on success, 84 on error

Definition at line 122 of file main.c.

```

123 {
124     parsing_info_t parsed_info;
125     server_t server;
126
127     if (ac <= 10 || check_mini_args(ac, av) == 1) {
128         display_help();
129         return 0;
130     }
131     if (setup_signal_handler() == -1)
132         server_err("Couldn't setup signal handler");
133     parse_args(ac, av, &parsed_info);
134     disp_args(&parsed_info);
135     create_server(&server, &parsed_info);
136     set_rdm_seed();
137     create_map(&server, &parsed_info);
138     init_server_eggs(&server);
139     server_loop(&server);
140     free_all(&server, &parsed_info);
141     close(server.s_fd);
142     return 0;
143 }
```

14.185.1.5 server_loop()

```
static void server_loop (
    server_t * server ) [static]
```

Definition at line 97 of file main.c.

```
98 {
99     while (should_continue_running()) {
100         if (server->should_run)
101             check_client(server);
102     }
103     printf("Server is shutting down\n");
104 }
```

14.185.1.6 set_rdm_seed()

```
static void set_rdm_seed (
    void ) [static]
```

Definition at line 106 of file main.c.

```
107 {
108     unsigned long seed = (unsigned long)time(NULL);
109
110     seed ^= (unsigned long)getpid();
111     seed ^= (unsigned long)(uintptr_t)&seed;
112     seed *= 0x5DEECE66ULL;
113     rand((unsigned int)(seed & 0xFFFFFFFF));
114 }
```

14.185.1.7 setup_signal_handler()

```
static int setup_signal_handler (
    void ) [static]
```

Definition at line 37 of file main.c.

```
38 {
39     struct sigaction sa_int;
40     struct sigaction sa_pipe;
41
42     memset(&sa_int, 0, sizeof(sa_int));
43     memset(&sa_pipe, 0, sizeof(sa_pipe));
44     sa_int.sa_handler = sigint_handler;
45     sigemptyset(&sa_int.sa_mask);
46     sa_int.sa_flags = 0;
47     if (sigaction(SIGINT, &sa_int, NULL) == -1) {
48         perror("Failed to setup SIGINT handler");
49         return -1;
50     }
51     sa_pipe.sa_handler = sigpipe_handler;
52     sigemptyset(&sa_pipe.sa_mask);
53     sa_pipe.sa_flags = 0;
54     if (sigaction(SIGPIPE, &sa_pipe, NULL) == -1) {
55         perror("Failed to setup SIGPIPE handler");
56         return -1;
57     }
58     return 0;
59 }
```

14.185.1.8 should_continue_running()

```
static int should_continue_running (
    void ) [static]
```

Definition at line 61 of file main.c.

```
62 {
63     int *keep_running = get_running_flag();
64
65     return *keep_running;
66 }
```

14.185.1.9 sigint_handler()

```
static void sigint_handler (
    int signum ) [static]
```

Definition at line 23 of file main.c.

```
24 {
25     int *flag = get_running_flag();
26 }
```

```

27     (void) signum;
28     *flag = 0;
29 }
```

14.185.1.10 sigpipe_handler()

```
static void sigpipe_handler (
    int signum ) [static]
```

Definition at line 31 of file main.c.

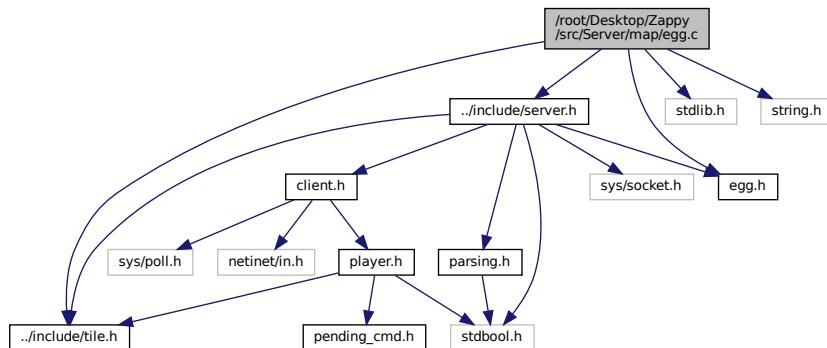
```

32 {
33     (void) signum;
34     perror("SIGPIPE Caught do not disconnect a client during function\n");
35 }
```

14.186 /root/Desktop/Zappy/src/Server/map/egg.c File Reference

```
#include "../include/tile.h"
#include "../include/server.h"
#include "../include/egg.h"
#include <stdlib.h>
#include <string.h>
```

Include dependency graph for egg.c:



Functions

- `egg_t * create_egg (int egg_id, int pos_x, int pos_y, char *team)`
- `static int remove_id_from_array (int **array, int *count, int id)`
- `static void remove_node (egg_t *current, egg_t *prev, server_t *server)`
- `static void remove_egg_by_id (server_t *server, int id)`
- `void remove_egg_from_tile (tile_t *tile, int egg_id)`
- `void remove_egg (server_t *server, int id, tile_t *tile)`
- `void add_egg_to_tile (tile_t *tile, int egg_id)`
- `void add_egg (server_t *server, egg_t *egg)`

14.186.1 Function Documentation

14.186.1.1 add_egg()

```
void add_egg (
    server_t * server,
    egg_t * egg )
```

Definition at line 114 of file egg.c.

```
115 {
116     tile_t *tile = NULL;
117
118     if (server == NULL || egg == NULL)
119         return;
120     egg->next = server->eggs;
121     server->eggs = egg;
122     tile = &server->map[egg->pos_y][egg->pos_x];
123     add_egg_to_tile(tile, egg->egg_id);
124 }
```

14.186.1.2 add_egg_to_tile()

```
void add_egg_to_tile (
    tile_t * tile,
    int egg_id )
```

Definition at line 92 of file egg.c.

```
93 {
94     int new_capacity;
95     int *new_egg_ids;
96
97     if (tile == NULL)
98         return;
99     if (tile->egg_count >= tile->egg_capacity) {
100         if (tile->egg_capacity == 0)
101             new_capacity = 4;
102         else
103             new_capacity = tile->egg_capacity * 2;
104         new_egg_ids = realloc(tile->egg_ids, sizeof(int) * new_capacity);
105         if (new_egg_ids == NULL)
106             return;
107         tile->egg_ids = new_egg_ids;
108         tile->egg_capacity = new_capacity;
109     }
110     tile->egg_ids[tile->egg_count] = egg_id;
111     tile->egg_count++;
112 }
```

14.186.1.3 create_egg()

```
egg_t* create_egg (
    int egg_id,
    int pos_x,
    int pos_y,
    char * team )
```

Definition at line 14 of file egg.c.

```
15 {
16     egg_t *new_egg = malloc(sizeof(egg_t));
17
18     if (new_egg == NULL)
19         return NULL;
20     new_egg->egg_id = egg_id;
21     new_egg->pos_x = pos_x;
22     new_egg->pos_y = pos_y;
23     new_egg->next = NULL;
24     new_egg->team_name = strdup(team);
25     if (new_egg->team_name == NULL)
26         server_err("Egg team allocation failed\n");
27     return new_egg;
28 }
```

14.186.1.4 remove_egg()

```
void remove_egg (
    server_t * server,
    int id,
    tile_t * tile )
```

Definition at line 84 of file egg.c.

```
85 {
86     if (tile == NULL)
```

```

87     return;
88     remove_egg_from_tile(tile, id);
89     remove_egg_by_id(server, id);
90 }

```

14.186.1.5 remove_egg_by_id()

```
static void remove_egg_by_id (
    server_t * server,
    int id ) [static]
```

Definition at line 62 of file egg.c.

```

63 {
64     egg_t *current = server->eggs;
65     egg_t *prev = NULL;
66
67     while (current != NULL) {
68         if (current->egg_id == id) {
69             remove_node(current, prev, server);
70             return;
71         }
72         prev = current;
73         current = current->next;
74     }
75 }
```

14.186.1.6 remove_egg_from_tile()

```
void remove_egg_from_tile (
    tile_t * tile,
    int egg_id )
```

Definition at line 77 of file egg.c.

```

78 {
79     if (tile == NULL)
80         return;
81     remove_id_from_array(&tile->egg_ids, &tile->egg_count, egg_id);
82 }
```

14.186.1.7 remove_id_from_array()

```
static int remove_id_from_array (
    int ** array,
    int * count,
    int id ) [static]
```

Definition at line 30 of file egg.c.

```

31 {
32     int last_idx = 0;
33
34     if (array == NULL || *array == NULL || count == NULL)
35         return -1;
36     for (int i = 0; i < *count; i++) {
37         if ((*array)[i] == id) {
38             last_idx = *count - 1;
39             (*array)[i] = (*array)[last_idx];
40             *count -= 1;
41             return 0;
42         }
43     }
44     return -1;
45 }
```

14.186.1.8 remove_node()

```
static void remove_node (
    egg_t * current,
    egg_t * prev,
    server_t * server ) [static]
```

Definition at line 47 of file egg.c.

```

48 {
49     egg_t *node_to_delete = NULL;
50
51     node_to_delete = current;
52     if (prev == NULL)
53         server->eggs = current->next;
54     else
55         prev->next = current->next;
56     if (node_to_delete->team_name)
57         free(node_to_delete->team_name);
58     free(node_to_delete);
59     return;
60 }

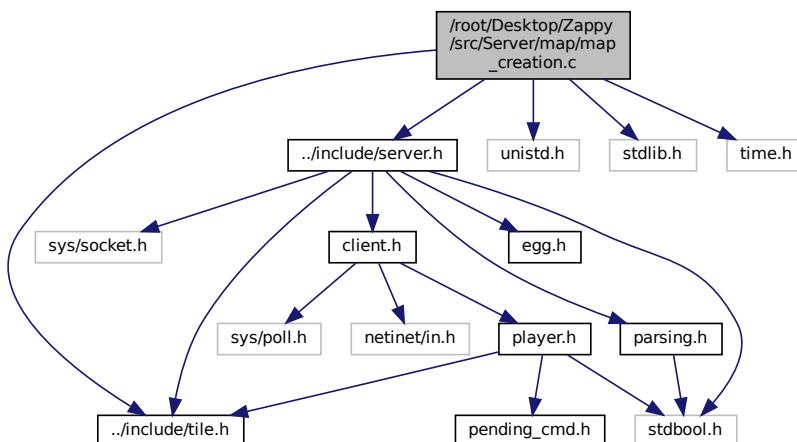
```

14.187 /root/Desktop/Zappy/src/Server/map/map_creation.c File Reference

```

#include "../include/tile.h"
#include "../include/server.h"
#include <unistd.h>
#include <stdlib.h>
#include <time.h>
Include dependency graph for map_creation.c:

```



Functions

- void **create_map** (server_t *server, parsing_info_t *parsed_info)
Initialise et génère la carte de jeu.

14.187.1 Function Documentation

14.187.1.1 create_map()

```

void create_map (
    server_t * server,
    parsing_info_t * parsed_info )

```

Initialise et génère la carte de jeu.

Parameters

<i>server</i>	Pointeur vers la structure du serveur
<i>parsed_info</i>	Informations de configuration pour la carte

Definition at line 13 of file map_creation.c.

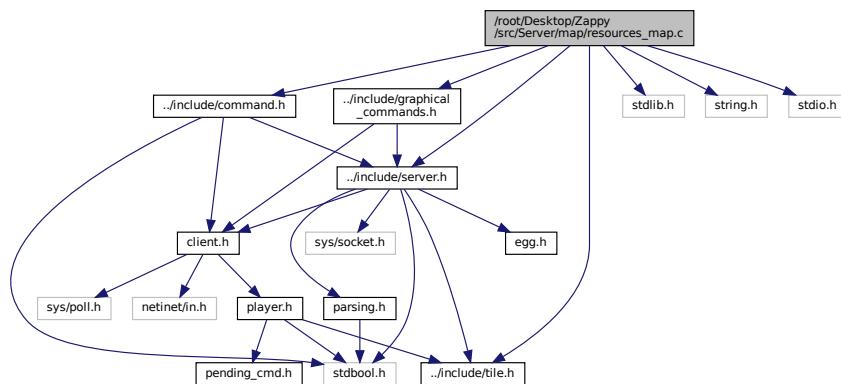
```

14 {
15     server->map = malloc(sizeof(tile_t *) * parsed_info->height);
16     if (server->map == NULL)
17         exit(84);
18     for (int y = 0; y < parsed_info->height; y++) {
19         server->map[y] = malloc(sizeof(tile_t) * parsed_info->width);
20         if (server->map[y] == NULL)
21             exit(84);
22         for (int x = 0; x < parsed_info->width; x++) {
23             tile_init(&server->map[y][x]);
24         }
25     }
26     distribute_resources(server->map, server,
27                         server->total_resources, server->current_resources);
28 }
```

14.188 /root/Desktop/Zappy/src/Server/map/resources_map.c File Reference

```
#include "../include/tile.h"
#include "../include/server.h"
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
```

Include dependency graph for resources_map.c:



Functions

- static void **shuffle_indices** (int *indices, int total_tiles)
- static void **distribute_one_resource** (resource_dist_t *dist, int res, int total, server_t *server)
- static void **init_resource_dist** (int width, int height, tile_t **map, resource_dist_t *dist)
- void **distribute_resources** (tile_t **map, server_t *server, int *total_resources, int *current_resources)
- void **respawn_resources** (tile_t **map, server_t *server, int *total_resources, int *current_resources)

14.188.1 Function Documentation

14.188.1.1 distribute_one_resource()

```
static void distribute_one_resource (
    resource_dist_t * dist,
    int res,
    int total,
    server_t * server ) [static]
```

Definition at line 29 of file resources_map.c.

```
31 {
32     int idx;
33     int y;
34     int x;
35
36     shuffle_indices(dist->tile_indices, dist->width * dist->height);
37     for (int i = 0; i < total; ++i) {
38         idx = dist->tile_indices[i];
39         y = idx / dist->width;
40         x = idx % dist->width;
41         dist->map[y][x].resources[res]++;
42         send_bct_to_all_graphical_clients(server, x, y);
43     }
44 }
```

14.188.1.2 distribute_resources()

```
void distribute_resources (
    tile_t ** map,
    server_t * server,
    int * total_resources,
    int * current_resources )
```

Definition at line 61 of file resources_map.c.

```
63 {
64     int total;
65     resource_dist_t dist;
66     static double resource_densities[COUNT] = {FOOD_D, LINEMATE_D, DERAUMERE_D,
67         SIBUR_D, MENDIANE_D, PHIRAS_D, THYSTAME_D};
68
69     init_resource_dist(server->parsed_info->width,
70         server->parsed_info->height, map, &dist);
71     for (int res = 0; res < COUNT; ++res) {
72         total = (int)(server->parsed_info->width *
73             server->parsed_info->height * resource_densities[res] + 0.5);
74         if (total < 1)
75             total = 1;
76         total_resources[res] = total;
77         current_resources[res] = total;
78         distribute_one_resource(&dist, res, total, server);
79     }
80     free(dist.tile_indices);
81 }
```

14.188.1.3 init_resource_dist()

```
static void init_resource_dist (
    int width,
    int height,
    tile_t ** map,
    resource_dist_t * dist ) [static]
```

Definition at line 46 of file resources_map.c.

```
48 {
49     int total_tiles = width * height;
50
51     dist->tile_indices = malloc(sizeof(int) * total_tiles);
52     if (dist->tile_indices == NULL)
53         exit(84);
54     for (int i = 0; i < total_tiles; ++i)
55         dist->tile_indices[i] = i;
56     dist->map = map;
57     dist->width = width;
58     dist->height = height;
59 }
```

14.188.1.4 respawn_resources()

```

void respawn_resources (
    tile_t ** map,
    server_t * server,
    int * total_resources,
    int * current_resources )

```

Definition at line 83 of file resources_map.c.

```

85 {
86     resource_dist_t dist;
87     int missing;
88
89     init_resource_dist(server->parsed_info->width,
90         server->parsed_info->height, map, &dist);
91     for (int res = 0; res < COUNT; ++res) {
92         missing = total_resources[res] - current_resources[res];
93         if (missing > 0) {
94             distribute_one_resource(&dist, res, missing, server);
95             current_resources[res] += missing;
96         }
97     }
98     send_smg_command(server, "Resources respawned");
99     free(dist.tile_indices);
100 }
```

14.188.1.5 shuffle_indices()

```

static void shuffle_indices (
    int * indices,
    int total_tiles ) [static]

```

Definition at line 16 of file resources_map.c.

```

17 {
18     int j;
19     int tmp;
20
21     for (int i = total_tiles - 1; i > 0; --i) {
22         j = rand() % (i + 1);
23         tmp = indices[i];
24         indices[i] = indices[j];
25         indices[j] = tmp;
26     }
27 }
```

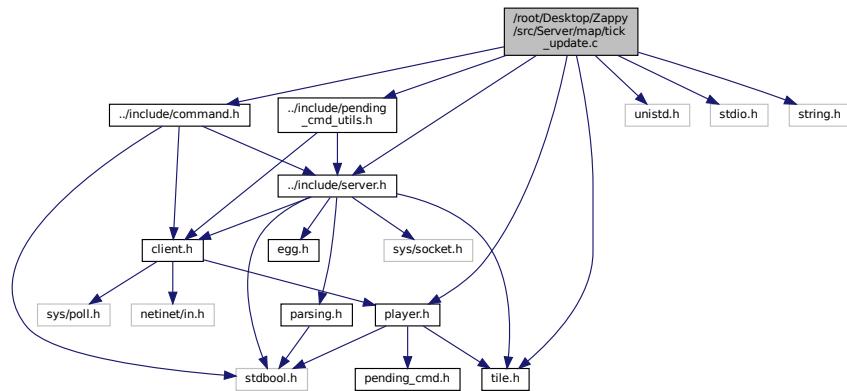
14.189 /root/Desktop/Zappy/src/Server/map/tick_update.c File Reference

```

#include "../include/server.h"
#include "../include/command.h"
#include "../include/player.h"
#include "../include/pending_cmd_utils.h"
#include "../include/tile.h"
#include <unistd.h>
#include <stdio.h>
#include <string.h>

```

Include dependency graph for tick_update.c:



Functions

- static bool [tick_check \(server_t *server, client_t *current\)](#)
- void [update_game_tick \(server_t *server\)](#)

Met à jour le tick de jeu et traite les actions en attente.

14.189.1 Function Documentation

14.189.1.1 tick_check()

```
static bool tick_check (
    server_t * server,
    client_t * current ) [static]
```

Definition at line 17 of file tick_update.c.

```
18 {
19     bool dead = false;
20
21     if (current != NULL && current->player != NULL)
22         dead = check_player_starvation(server, current);
23     if (dead == false && current != NULL &&
24         current->player != NULL && current->player->is_in_incantation)
25         finish_incantation(server, current);
26     return dead;
27 }
```

14.189.1.2 update_game_tick()

```
void update_game_tick (
    server_t * server )
```

Met à jour le tick de jeu et traite les actions en attente.

Parameters

<code>server</code>	Pointeur vers la structure du serveur
---------------------	---------------------------------------

Definition at line 29 of file tick_update.c.

```
30 {
31     client_t *current = server->client;
32     client_t *next;
33
34     server->current_tick++;
35     if (server->current_tick % 20 == 0)
```

```

36     respawn_resources(server->map, server, server->total_resources,
37                         server->current_resources);
38     if (current != NULL)
39         current = current->next;
40     while (current != NULL) {
41         next = current->next;
42         if (current->player && current->player->pending_cmd &&
43             current->player->busy_until <= server->current_tick)
44             execute_pending_cmd(server, current);
45         if (tick_check(server, current) == false && current->player &&
46             current->player->busy_until <= server->current_tick &&
47             current->player->queue_size > 0)
48             process_next_queued_command(server, current);
49         current = next;
50     }
51 }

```

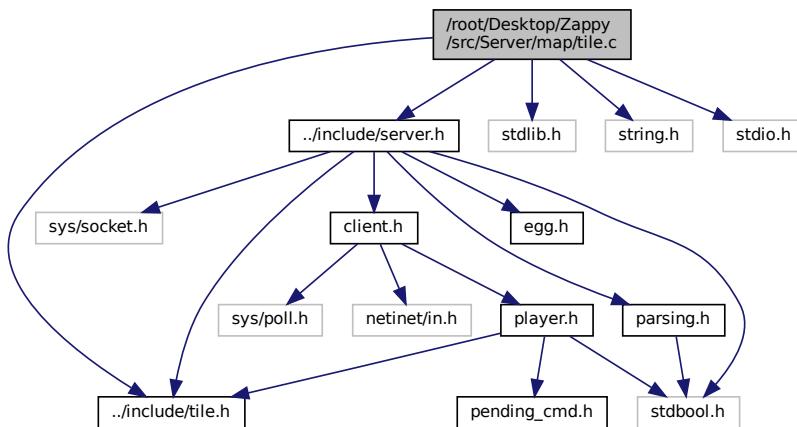
14.190 /root/Desktop/Zappy/src/Server/map/tile.c File Reference

```

#include "../include/tile.h"
#include "../include/server.h"
#include <stdlib.h>
#include <string.h>
#include <stdio.h>

```

Include dependency graph for tile.c:



Functions

- void `tile_init (tile_t *tile)`
- void `tile_free (tile_t *tile)`
- int `tile_add_player (tile_t *tile, int player_id)`
- int `tile_remove_player (tile_t *tile, int player_id)`

14.190.1 Function Documentation

14.190.1.1 tile_add_player()

```

int tile_add_player (
    tile_t * tile,
    int player_id )

```

Definition at line 37 of file tile.c.

```

38 {
39     int new_capacity;
40     int *new_ids;
41
42     if (tile->player_count >= tile->player_capacity) {
43         if (tile->player_capacity == 0)
44             new_capacity = 4;
45         else
46             new_capacity = tile->player_capacity * 2;
47         new_ids = realloc(tile->player_ids, new_capacity * sizeof(int));
48         if (!new_ids)
49             return -1;
50         tile->player_ids = new_ids;
51         tile->player_capacity = new_capacity;
52     }
53     tile->player_ids[tile->player_count] = player_id;
54     tile->player_count++;
55     return 0;
56 }

```

14.190.1.2 tile_free()

```
void tile_free (
    tile_t * tile )
```

Definition at line 25 of file tile.c.

```

26 {
27     free(tile->player_ids);
28     tile->player_ids = NULL;
29     tile->player_count = 0;
30     tile->player_capacity = 0;
31     free(tile->egg_ids);
32     tile->egg_ids = NULL;
33     tile->egg_count = 0;
34     tile->egg_capacity = 0;
35 }
```

14.190.1.3 tile_init()

```
void tile_init (
    tile_t * tile )
```

Definition at line 14 of file tile.c.

```

15 {
16     tile->player_ids = NULL;
17     tile->player_count = 0;
18     tile->player_capacity = 0;
19     tile->egg_ids = NULL;
20     tile->egg_count = 0;
21     tile->egg_capacity = 0;
22     memset(tile->resources, 0, sizeof(tile->resources));
23 }
```

14.190.1.4 tile_remove_player()

```
int tile_remove_player (
    tile_t * tile,
    int player_id )
```

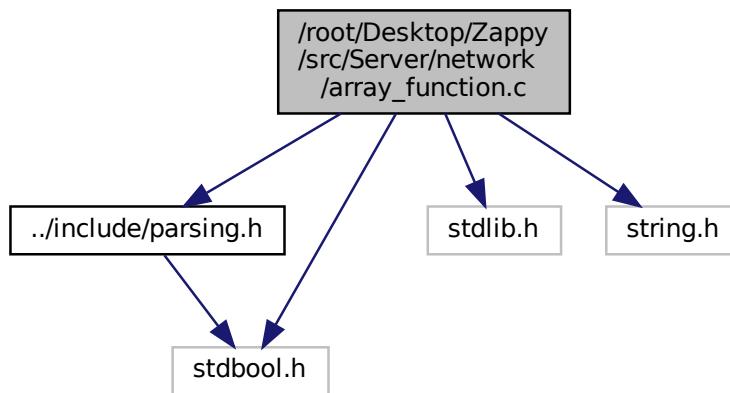
Definition at line 58 of file tile.c.

```

59 {
60     for (int i = 0; i < tile->player_count; ++i) {
61         if (tile->player_ids[i] == player_id) {
62             tile->player_ids[i] = tile->player_ids[tile->player_count - 1];
63             tile->player_count--;
64         }
65     }
66 }
67 return -1;
68 }
```

14.191 /root/Desktop/Zappy/src/Server/network/array_function.c File Reference

```
#include "../include/parsing.h"
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
Include dependency graph for array_function.c:
```



Functions

- int [arr_len](#) (char **array)
- static char * [next_valid_token](#) (char *str, char *delim, int *first)
- static int [count_valid_tokens](#) (char *str, char *delim)
- static char ** [init_result_array](#) (int size)
- static char ** [free_cpy_arr](#) (char *copy, char **array)
- static int [add_token](#) (char **array, char *token, int pos)
- static bool [check_if_failed](#) (char *copy, char **array)
- static char ** [fill_result_array](#) (char *str, char *delim, int size)
- static char ** [process_tokens](#) (char *str_copy, char *delim)
- char ** [str_to_word_arr](#) (char *str, char *delim)

14.191.1 Function Documentation

14.191.1.1 add_token()

```
static int add_token (
    char ** array,
    char * token,
    int pos ) [static]
Definition at line 81 of file array_function.c.
82 {
83     array[pos] = strdup(token);
84     if (array[pos] == NULL)
85         return 0;
86     return 1;
87 }
```

14.191.1.2 arr_len()

```
int arr_len (
    char ** array )
Definition at line 13 of file array_function.c.
14 {
15     int i = 0;
16
17     if (!array)
18         return 0;
19     while (array[i])
20         i++;
21     return i;
22 }
```

14.191.1.3 check_if_failed()

```
static bool check_if_failed (
    char * copy,
    char ** array ) [static]
Definition at line 89 of file array_function.c.
90 {
91     if (!copy) {
92         if (array)
93             free_arr(array);
94         return true;
95     }
96     if (!array) {
97         free(copy);
98         return true;
99     }
100    return false;
101 }
```

14.191.1.4 count_valid_tokens()

```
static int count_valid_tokens (
    char * str,
    char * delim ) [static]
Definition at line 41 of file array_function.c.
42 {
43     char *copy = strdup(str);
44     char *token;
45     int count = 0;
46     int first = 0;
47
48     if (!copy)
49         return -1;
50     while (1) {
51         token = next_valid_token(copy, delim, &first);
52         if (!token)
53             break;
54         count++;
55     }
56     if (token)
57         free(token);
58     free(copy);
59     return count;
60 }
```

14.191.1.5 fill_result_array()

```
static char** fill_result_array (
    char * str,
    char * delim,
    int size ) [static]
Definition at line 103 of file array_function.c.
104 {
105     char *copy = strdup(str);
106     char **array = init_result_array(size);
107     char *token;
```

```

108     int pos = 0;
109     int first = 0;
110
111     if (check_if_failed(copy, array))
112         return NULL;
113     while (1) {
114         token = next_valid_token(copy, delim, &first);
115         if (!token || pos >= size)
116             break;
117         if (!add_token(array, token, pos))
118             return free_cpy_arr(copy, array);
119         pos++;
120     }
121     if (token)
122         free(token);
123     free(copy);
124     return array;
125 }
```

14.191.1.6 free_cpy_arr()

```

static char** free_cpy_arr (
    char * copy,
    char ** array ) [static]
```

Definition at line 72 of file array_function.c.

```

73 {
74     if (array)
75         free_arr(array);
76     if (copy)
77         free(copy);
78     return NULL;
79 }
```

14.191.1.7 init_result_array()

```

static char** init_result_array (
    int size ) [static]
```

Definition at line 62 of file array_function.c.

```

63 {
64     char **array = malloc(sizeof(char *) * (size + 1));
65
66     if (!array)
67         return NULL;
68     array[size] = NULL;
69     return array;
70 }
```

14.191.1.8 next_valid_token()

```

static char* next_valid_token (
    char * str,
    char * delim,
    int * first ) [static]
```

Definition at line 24 of file array_function.c.

```

25 {
26     char *token;
27
28     while (1) {
29         if (*first == 0) {
30             token = strtok(str, delim);
31             *first = 1;
32         } else
33             token = strtok(NULL, delim);
34         if (token == NULL)
35             return NULL;
36         if (strlen(token) > 0)
37             return token;
38     }
39 }
```

14.191.1.9 process_tokens()

```
static char** process_tokens (
    char * str_copy,
    char * delim ) [static]
Definition at line 127 of file array_function.c.
128 {
129     int count;
130     char **array;
131
132     count = count_valid_tokens(str_copy, delim);
133     if (count < 0)
134         return NULL;
135     if (count == 0) {
136         array = malloc(sizeof(char *));
137         if (!array)
138             return NULL;
139         array[0] = NULL;
140         return array;
141     }
142     array = fill_result_array(str_copy, delim, count);
143     return array;
144 }
```

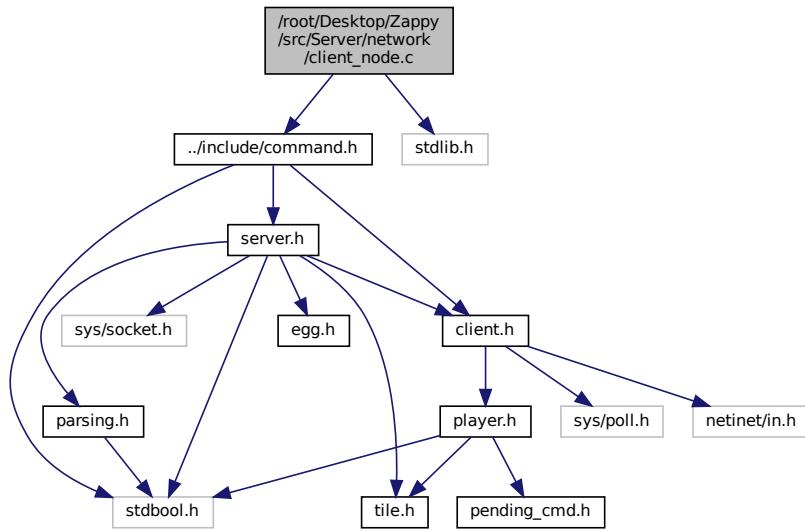
14.191.1.10 str_to_word_arr()

```
char** str_to_word_arr (
    char * str,
    char * delim )
Definition at line 146 of file array_function.c.
147 {
148     char *str_copy;
149     char **result;
150     int len = 0;
151
152     if (!str || !delim)
153         return NULL;
154     len = strlen(str);
155     str_copy = strdup(str);
156     if (!str_copy)
157         return NULL;
158     if (len > 0 && str[len - 1] == '\n')
159         str_copy[len - 1] = '\0';
160     if (!str_copy)
161         return NULL;
162     result = process_tokens(str_copy, delim);
163     free(str_copy);
164     return result;
165 }
```

14.192 /root/Desktop/Zappy/src/Server/network/client_node.c File Reference

```
#include "../include/command.h"
#include <stdlib.h>
```

Include dependency graph for client_node.c:



Functions

- static void `cleanup_player_client (client_t *current_client)`
- void `free_node (client_t *node, server_t *server)`

Libère la mémoire d'un nud client et le supprime du serveur.

14.192.1 Function Documentation

14.192.1.1 cleanup_player_client()

```
static void cleanup_player_client (
    client_t * current_client ) [static]
```

Definition at line 11 of file client_node.c.

```

12 {
13     player_t *player = current_client->player;
14
15     free(player->team_name);
16     player->team_name = NULL;
17     free_inventory(player);
18     if (player->command_queue)
19         cleanup_player_queue(player);
20     if (player->pending_cmd) {
21         if (player->pending_cmd->args)
22             free(player->pending_cmd->args);
23         player->pending_cmd->args = NULL;
24     }
25     free(player->pending_cmd);
26     player->pending_cmd = NULL;
27 }
28 free(player);
29 current_client->player = NULL;
30 }
```

14.192.1.2 free_node()

```
void free_node (
    client_t * node,
    server_t * server )
```

Libère la mémoire d'un nud client et le supprime du serveur.

Parameters

<i>node</i>	Nud client à supprimer
<i>server</i>	Pointeur vers la structure du serveur

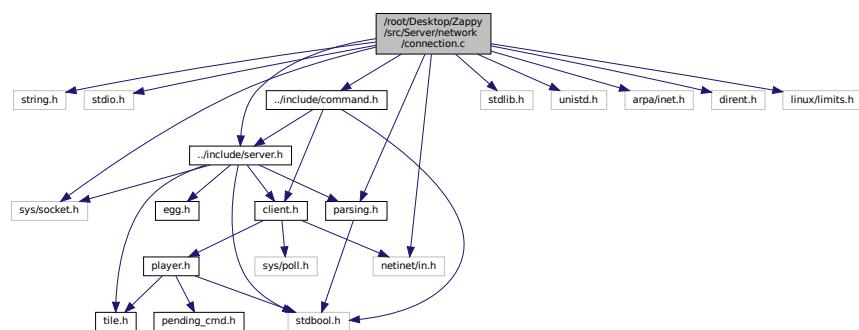
Definition at line 32 of file client_node.c.

```

33 {
34     if (node->type == GRAPHICAL)
35         remove_graphic_client(server, node);
36     if (node->client_poll)
37         free(node->client_poll);
38     if (node->client_add)
39         free(node->client_add);
40     if (node->player) {
41         cleanup_player_client(node);
42         cleanup_pending(node->player);
43         free(node->player);
44     }
45     free(node);
46 }
```

14.193 /root/Desktop/Zappy/src/Server/network/connection.c File Reference

```
#include <string.h>
#include <stdio.h>
#include <sys/socket.h>
#include <stdlib.h>
#include <unistd.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <dirent.h>
#include <linux/limits.h>
#include "../include/server.h"
#include "../include/parsing.h"
#include "../include/command.h"
Include dependency graph for connection.c:
```



Functions

- static bool `remove_head_client (server_t *server, int fd)`
- static bool `remove_other_client (server_t *server, int fd)`
- void `remove_fd (server_t *server, int fd)`

Supprime un descripteur de fichier du gestionnaire de polling.

- static `client_t * init_new_client (int fd)`
- void `add_fd (server_t *server, int fd)`
Ajoute un descripteur de fichier au gestionnaire de polling.
- static void `init_server_socket (server_t *server, parsing_info_t *parsed_info)`
- static void `copy_names (server_t *server, parsing_info_t *parsed_info)`
- void `init_server_resources (server_t *server)`
- static void `init_server (server_t *server, parsing_info_t *parsed_info)`
- void `create_server (server_t *server, parsing_info_t *parsed_info)`
Initialise et configure le serveur.

14.193.1 Function Documentation

14.193.1.1 add_fd()

```
void add_fd (
    server_t * server,
    int fd )
```

Ajoute un descripteur de fichier au gestionnaire de polling.

Parameters

<code>server</code>	Pointeur vers la structure du serveur
<code>fd</code>	Descripteur de fichier à ajouter

Definition at line 95 of file connection.c.

```
96 {
97     static int next_id = 0;
98     client_t *new_c = init_new_client(fd);
99     client_t *current;
100
101    if (server->client == NULL) {
102        server->client = new_c;
103        new_c->client_id = -1;
104        return;
105    }
106    current = server->client;
107    while (current->next != NULL) {
108        current = current->next;
109    }
110    new_c->client_id = next_id;
111    next_id++;
112    current->next = new_c;
113 }
```

14.193.1.2 copy_names()

```
static void copy_names (
    server_t * server,
    parsing_info_t * parsed_info ) [static]
```

Definition at line 138 of file connection.c.

```
139 {
140     int i = 0;
141
142     for (; parsed_info->names[i] != NULL; i++) {
143     }
144     server->parsed_info->names = calloc(i + 1, sizeof(char *));
145     if (server->parsed_info->names == NULL)
146         server_err("Malloc failed for parsing info names copy into server");
147     for (int j = 0; parsed_info->names[j] != NULL; j++) {
148         server->parsed_info->names[j] = strdup(parsed_info->names[j]);
149     }
150     server->parsed_info->names[i] = NULL;
151 }
```

14.193.1.3 create_server()

```
void create_server (
    server_t * server,
    parsing_info_t * parsed_info )
```

Initialise et configure le serveur.

Parameters

<i>server</i>	Pointeur vers la structure du serveur à initialiser
<i>parsed_info</i>	Informations de configuration parsées

Definition at line 189 of file connection.c.

```
190 {
191     init_server(server, parsed_info);
192     init_server_socket(server, parsed_info);
193 }
```

14.193.1.4 init_new_client()

```
static client_t* init_new_client (
    int fd ) [static]
```

Definition at line 71 of file connection.c.

```
72 {
73     client_t *new_c = calloc(1, sizeof(client_t));
74
75     if (!new_c)
76         server_err("New client allocation failed");
77     new_c->client_poll = calloc(1, sizeof(struct pollfd));
78     if (!new_c->client_poll)
79         server_err("Poll fd struct allocation in new client struct failed");
80     new_c->client_poll->fd = fd;
81     new_c->client_poll->events = POLLIN;
82     new_c->client_poll->revents = 0;
83     new_c->next = NULL;
84     new_c->client_fd = fd;
85     new_c->client_addr = NULL;
86     new_c->client_id = -1;
87     new_c->player = calloc(1, sizeof(player_t));
88     new_c->is_fully_connected = false;
89     if (new_c->player == NULL)
90         server_err("Failed to allocate player");
91     init_player(new_c->player, NULL);
92     return new_c;
93 }
```

14.193.1.5 init_server()

```
static void init_server (
    server_t * server,
    parsing_info_t * parsed_info ) [static]
```

Definition at line 165 of file connection.c.

```
166 {
167     server->nfds = 0;
168     server->client = NULL;
169     server->graphical_clients = NULL;
170     server->s_fd = 0;
171     server->serv_addr = NULL;
172     server->current_tick = 0;
173     server->map = NULL;
174     server->parsed_info = malloc(sizeof(parsing_info_t));
175     if (server->parsed_info == NULL)
176         server_err("Malloc failed for allocating parsed_info in server");
177     server->parsed_info->port = parsed_info->port;
178     server->parsed_info->width = parsed_info->width;
179     server->parsed_info->height = parsed_info->height;
180     server->parsed_info->client_nb = parsed_info->client_nb;
181     server->parsed_info->frequence = parsed_info->frequence;
182     server->eggs = NULL;
183     server->should_run = true;
184     server->poll_manager = calloc(1, sizeof(poll_manager_t));
185     init_server_resources(server);
```

```
186     copy_names(server, parsed_info);
187 }
```

14.193.1.6 init_server_resources()

```
void init_server_resources (
    server_t * server )
```

Definition at line 153 of file connection.c.

```
154 {
155     server->total_resources = malloc(sizeof(int) * COUNT);
156     server->current_resources = malloc(sizeof(int) * COUNT);
157     if (server->total_resources == NULL || server->current_resources == NULL)
158         server_err("Malloc failed for allocating resource counters");
159     for (int i = 0; i < COUNT; i++) {
160         server->total_resources[i] = 0;
161         server->current_resources[i] = 0;
162     }
163 }
```

14.193.1.7 init_server_socket()

```
static void init_server_socket (
    server_t * server,
    parsing_info_t * parsed_info ) [static]
```

Definition at line 115 of file connection.c.

```
116 {
117     int opt = 1;
118
119     server->s_fd = socket(AF_INET, SOCK_STREAM, 0);
120     if (server->s_fd == -1)
121         server_err("Socket creation failed");
122     if (setsockopt(server->s_fd, SOL_SOCKET, SO_REUSEADDR, &opt, sizeof(opt)))
123         server_err("setsockopt failed");
124     server->serv_addr = calloc(1, sizeof(struct sockaddr_in));
125     if (server->serv_addr == NULL)
126         server_err("Failed to allocate server address");
127     server->serv_addr->sin_family = AF_INET;
128     server->serv_addr->sin_addr.s_addr = INADDR_ANY;
129     server->serv_addr->sin_port = htons(parsed_info->port);
130     if (bind(server->s_fd, (struct sockaddr *)server->serv_addr,
131             sizeof(struct sockaddr_in)) < 0)
132         server_err("Connection bind failed");
133     if (listen(server->s_fd, 1000) < 0)
134         server_err("Connection listen failed");
135     add_fd(server, server->s_fd);
136 }
```

14.193.1.8 remove_fd()

```
void remove_fd (
    server_t * server,
    int fd )
```

Supprime un descripteur de fichier du gestionnaire de polling.

Parameters

<i>server</i>	Pointeur vers la structure du serveur
<i>fd</i>	Descripteur de fichier à supprimer

Definition at line 61 of file connection.c.

```
62 {
63     if (server->client == NULL)
64         return;
65     if (remove_head_client(server, fd))
66         return;
67     remove_other_client(server, fd);
68     close(fd);
69 }
```

14.193.1.9 remove_head_client()

```
static bool remove_head_client (
    server_t * server,
    int fd ) [static]
```

Definition at line 20 of file connection.c.

```
21 {
22     client_t *current = server->client;
23     int client_id;
24
25     if (current->client_fd != fd)
26         return false;
27     client_id = current->client_id;
28     if (current->client_fd == server->s_fd) {
29         printf("Error: Attempting to remove server listening socket\n");
30         return true;
31     }
32     server->client = current->next;
33     free_node(current, server);
34     server->nfds -= 1;
35     printf("Client %d disconnected\n", client_id);
36     return true;
37 }
```

14.193.1.10 remove_other_client()

```
static bool remove_other_client (
    server_t * server,
    int fd ) [static]
```

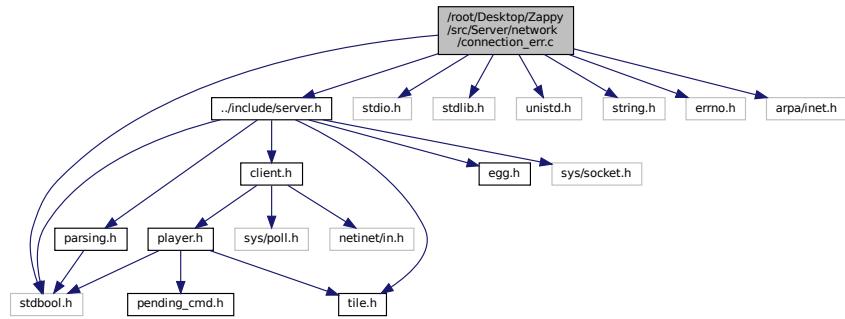
Definition at line 39 of file connection.c.

```
40 {
41     client_t *current = server->client;
42     client_t *prev = NULL;
43     int client_id;
44
45     while (current != NULL && current->client_fd != fd) {
46         prev = current;
47         current = current->next;
48     }
49     if (current == NULL) {
50         printf("Error: Client with fd %d not found during removal\n", fd);
51         return false;
52     }
53     client_id = current->client_id;
54     prev->next = current->next;
55     free_node(current, server);
56     server->nfds -= 1;
57     printf("Client %d disconnected\n", client_id);
58     return true;
59 }
```

14.194 /root/Desktop/Zappy/src/Server/network/connection_err.c File Reference

```
#include "../include/server.h"
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <stdbool.h>
#include <string.h>
#include <errno.h>
#include <arpa/inet.h>
```

Include dependency graph for connection_err.c:



Functions

- void `server_err` (char *msg)
Affiche un message d'erreur et termine le programme.
- void `print_co` (char *client_ip, struct sockaddr_in *client_addr, `client_t` *new_client)
Affiche les informations de connexion d'un nouveau client.

14.194.1 Function Documentation

14.194.1.1 print_co()

```
void print_co (
    char * client_ip,
    struct sockaddr_in * client_addr,
    client_t * new_client )
```

Affiche les informations de connexion d'un nouveau client.

Parameters

<code>client_ip</code>	Adresse IP du client
<code>client_addr</code>	Structure d'adresse du client
<code>new_client</code>	Structure du nouveau client

Definition at line 23 of file connection_err.c.

```
25 {
26     inet_ntop(AF_INET, &(client_addr->sin_addr), client_ip, INET_ADDRSTRLEN);
27     printf("New client connected from %s:%d with ID %d.\n",
28            client_ip, ntohs(client_addr->sin_port), new_client->client_id);
29 }
```

14.194.1.2 server_err()

```
void server_err (
    char * msg )
```

Affiche un message d'erreur et termine le programme.

Parameters

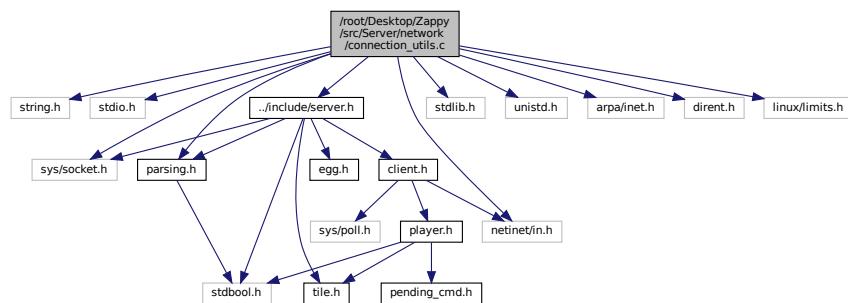
<code>msg</code>	Message d'erreur à afficher
------------------	-----------------------------

Definition at line 17 of file connection_err.c.

```
18 {
19     perror(msg);
20     exit(84);
21 }
```

14.195 /root/Desktop/Zappy/src/Server/network/connection_utils.c File Reference

```
#include <string.h>
#include <stdio.h>
#include <sys/socket.h>
#include <stdlib.h>
#include <unistd.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <dirent.h>
#include <linux/limits.h>
#include "../include/server.h"
#include "../include/parsing.h"
Include dependency graph for connection_utils.c:
```



Functions

- int [count_team \(server_t *n_server\)](#)
Compte le nombre d'équipes configurées.
- static void [create_server_egg \(server_t *n_server, int egg_id, int i\)](#)
- void [init_server_eggs \(server_t *n_server\)](#)
Initialise les ufs de départ sur le serveur.

14.195.1 Function Documentation

14.195.1.1 count_team()

```
int count_team (
    server_t * n_server )
Compte le nombre d'équipes configurées.
```

Parameters

<i>n_server</i>	Pointeur vers la structure du serveur
-----------------	---------------------------------------

Returns

Nombre d'équipes

Definition at line 19 of file connection_utils.c.

```
20 {
21     int i = 0;
22
23     for (; n_server->parsed_info->names[i]; i++) {
24     }
25     return i;
26 }
```

14.195.1.2 create_server_egg()

```
static void create_server_egg (
    server_t * n_server,
    int egg_id,
    int i ) [static]
```

Definition at line 28 of file connection_utils.c.

```
29 {
30     int pos_x = rand() % n_server->parsed_info->width;
31     int pos_y = rand() % n_server->parsed_info->height;
32     egg_t *n_egg = create_egg(egg_id, pos_x, pos_y,
33         n_server->parsed_info->names[i]);
34
35     if (n_egg == NULL)
36         server_err("Error in egg creation");
37     add_egg(n_server, n_egg);
38 }
```

14.195.1.3 init_server_eggs()

```
void init_server_eggs (
    server_t * n_server )
```

Initialise les ufs de départ sur le serveur.

Parameters

<i>n_server</i>	Pointeur vers la structure du serveur
-----------------	---------------------------------------

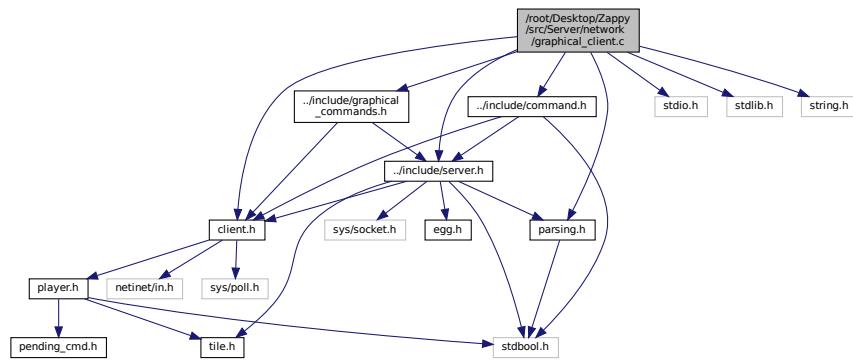
Definition at line 40 of file connection_utils.c.

```
41 {
42     int team_count = count_team(n_server);
43     int eggs_per_team = n_server->parsed_info->client_nb;
44     int egg_id = 0;
45
46     for (int i = 0; i < team_count; i++) {
47         for (int j = 0; j < eggs_per_team; j++) {
48             create_server_egg(n_server, egg_id, i);
49             egg_id++;
50         }
51     }
52 }
```

14.196 /root/Desktop/Zappy/src/Server/network/graphical_client.c File Reference

```
#include "../include/server.h"
#include "../include/client.h"
#include "../include/command.h"
#include "../include/graphical_commands.h"
#include "../include/parsing.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Include dependency graph for graphical_client.c:



Functions

- static void [add_to_graphical_list](#) (server_t *server, graphical_client_t *new)
- void [add_graphic_client](#) (server_t *server, client_t *client)
- static bool [remove_graphical_node](#) (server_t *server, client_t *client, graphical_client_t **current, graphical_client_t **prev)
- void [remove_graphic_client](#) (server_t *server, client_t *client)
- void [send_message_to_all_graphic](#) (server_t *server, char *message)
- void [send_map_info_to_one_client](#) (server_t *server, client_t *client)

14.196.1 Function Documentation

14.196.1.1 add_graphic_client()

```
void add_graphic_client (
    server_t * server,
    client_t * client )
```

Definition at line 32 of file graphical_client.c.

```
33 {
34     graphical_client_t *new_ref = malloc(sizeof(graphical_client_t));
35
36     if (!new_ref)
37         return;
38     new_ref->client = client;
39     new_ref->next = server->graphical_clients;
40     add_to_graphical_list(server, new_ref);
41 }
```

14.196.1.2 add_to_graphical_list()

```
static void add_to_graphical_list (
    server_t * server,
    graphical_client_t * new ) [static]
```

Definition at line 17 of file graphical_client.c.

```
18 {
19     graphical_client_t *current = NULL;
20
21     if (server->graphical_clients == NULL) {
22         server->graphical_clients = new;
23     } else {
24         current = server->graphical_clients;
25         while (current->next != NULL)
26             current = current->next;
27         current->next = new;
```

```

28     }
29     new->next = NULL;
30 }

```

14.196.1.3 remove_graphic_client()

```

void remove_graphic_client (
    server_t * server,
    client_t * client )

```

Definition at line 59 of file graphical_client.c.

```

60 {
61     graphical_client_t *current = server->graphical_clients;
62     graphical_client_t *prev = NULL;
63
64     if (!server || !client || !server->graphical_clients)
65         return;
66     while (current != NULL) {
67         if (remove_graphical_node(server, client, &current, &prev))
68             return;
69     }
70     remove_fd(server, client->client_fd);
71 }

```

14.196.1.4 remove_graphical_node()

```

static bool remove_graphical_node (
    server_t * server,
    client_t * client,
    graphical_client_t ** current,
    graphical_client_t ** prev ) [static]

```

Definition at line 43 of file graphical_client.c.

```

45 {
46     if ((*current)->client == client) {
47         if (*prev == NULL)
48             server->graphical_clients = (*current)->next;
49         else
50             (*prev)->next = (*current)->next;
51         free(*current);
52         return true;
53     }
54     *prev = *current;
55     *current = (*current)->next;
56     return false;
57 }

```

14.196.1.5 send_map_info_to_one_client()

```

void send_map_info_to_one_client (
    server_t * server,
    client_t * client )

```

Definition at line 84 of file graphical_client.c.

```

85 {
86     char **tmp;
87
88     if (!server || !client)
89         return;
90     tmp = malloc(sizeof(char *) * 2);
91     if (!tmp)
92         return;
93     tmp[0] = strdup("sgt\n");
94     tmp[1] = NULL;
95     send_msz_command(server, client);
96     command_sgt(server, client, tmp);
97     send_tile_content_to_one_client(server, client);
98     send_tna_command(server, client);
99     send_all_player_info_to_one_client(server, client);
100    send_enw_command_start(server);
101    free_arr(tmp);
102 }

```

14.196.1.6 send_message_to_all_graphic()

```
void send_message_to_all_graphic (
    server_t * server,
    char * message )
```

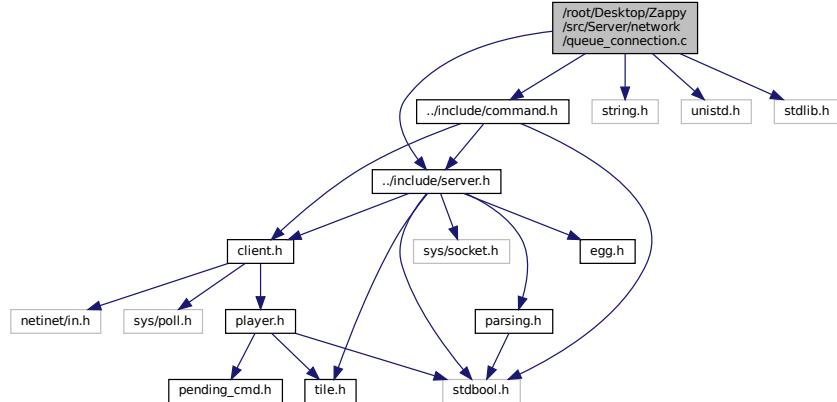
Definition at line 73 of file graphical_client.c.

```
74 {
75     graphical_client_t *current = server->graphical_clients;
76
77     while (current) {
78         if (current->client && current->client->client_fd != -1)
79             write_command_output(current->client->client_fd, message);
80         current = current->next;
81     }
82 }
```

14.197 /root/Desktop/Zappy/src/Server/network/queue_connection.c File Reference

```
#include "../include/server.h"
#include "../include/command.h"
#include <string.h>
#include <unistd.h>
#include <stdlib.h>
```

Include dependency graph for queue_connection.c:



Functions

- void [add_to_command_queue \(server_t *server, client_t *client, char *command\)](#)
Ajoute une commande à la file d'attente d'un client.
- static void [check_rebuild \(bool was_full, client_t *client, server_t *server\)](#)
- void [process_next_queued_command \(server_t *server, client_t *client\)](#)
Traite la prochaine commande en file d'attente pour un client.

14.197.1 Function Documentation

14.197.1.1 add_to_command_queue()

```
void add_to_command_queue (
    server_t * server,
```

```
    client_t * client,
    char * command )
```

Ajoute une commande à la file d'attente d'un client.

Parameters

<i>server</i>	Pointeur vers la structure du serveur
<i>client</i>	Client destinataire de la commande
<i>command</i>	Commande à ajouter à la file

Definition at line 14 of file queue_connection.c.

```
15 {
16     if (!client || !client->player || !command ||
17         client->player->queue_size >= 10 || !client->player->command_queue)
18         return;
19     client->player->command_queue[client->player->queue_size] =
20         strdup(command);
21     client->player->queue_size++;
22     if (client->player->queue_size >= 10 && server && server->poll_manager)
23         server->poll_manager->needs_rebuild = true;
24 }
```

14.197.1.2 check_rebuild()

```
static void check_rebuild (
    bool was_full,
    client_t * client,
    server_t * server ) [static]
```

Definition at line 26 of file queue_connection.c.

```
27 {
28     if (was_full && client->player->queue_size < 10 &&
29         server && server->poll_manager)
30         server->poll_manager->needs_rebuild = true;
31 }
```

14.197.1.3 process_next_queued_command()

```
void process_next_queued_command (
    server_t * server,
    client_t * client )
```

Traite la prochaine commande en file d'attente pour un client.

Parameters

<i>server</i>	Pointeur vers la structure du serveur
<i>client</i>	Client dont la commande doit être traitée

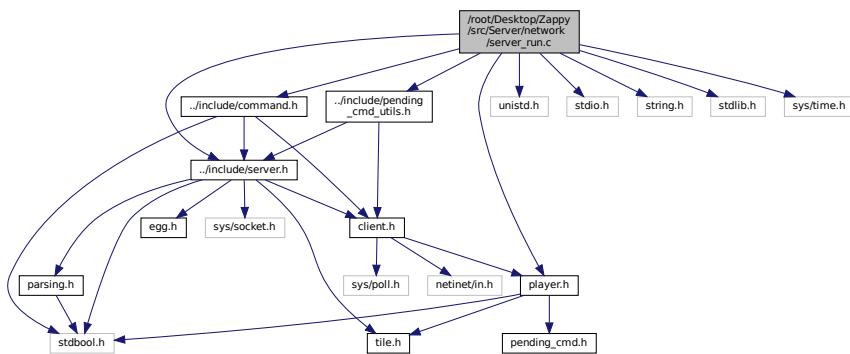
Definition at line 33 of file queue_connection.c.

```
34 {
35     char *next_command;
36     bool was_full;
37
38     if (!client || !client->player || !server ||
39         client->player->queue_size <= 0 || !client->player->command_queue)
40         return;
41     was_full = (client->player->queue_size >= 10);
42     next_command = client->player->command_queue[0];
43     for (int i = 0; i < client->player->queue_size - 1; i++) {
44         client->player->command_queue[i] =
45             client->player->command_queue[i + 1];
46     }
47     client->player->command_queue[client->player->queue_size - 1] = NULL;
48     client->player->queue_size--;
49     check_rebuild(was_full, client, server);
50     if (next_command) {
51         execute_com(server, client, next_command);
52         free(next_command);
53         next_command = NULL;
```

```
54     }
55 }
```

14.198 /root/Desktop/Zappy/src/Server/network/server_run.c File Reference

```
#include "../include/server.h"
#include "../include/command.h"
#include "../include/player.h"
#include "../include/pending_cmd_utils.h"
#include <unistd.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <sys/time.h>
Include dependency graph for server_run.c:
```



Functions

- `client_t * find_client_by_socket (server_t *server, int socket_fd)`
- `static void new_connection (server_t *server)`
- `static void check_new_connection (server_t *server)`
- `static void check_client_message (server_t *server)`
- `static void setup_poll_manager (poll_manager_t *poll_mana, int size)`
- `static void smart_polling (client_t *current, poll_manager_t *poll_mana, server_t *server, int i)`
- `static void fill_poll_array (server_t *server, poll_manager_t *poll_mana)`
- `static void poll_client (server_t *server, poll_manager_t *poll_mana)`
- `static void handle_game_tick (server_t *server)`
- `void check_client (server_t *server)`

Vérifie et traite les nouvelles connexions clients.

14.198.1 Function Documentation

14.198.1.1 check_client()

```
void check_client (
    server_t * server )
```

Vérifie et traite les nouvelles connexions clients.

Parameters

<code>server</code>	Pointeur vers la structure du serveur
---------------------	---------------------------------------

Definition at line 151 of file server_run.c.

```
152 {
153     int size = server->nfds + 1;
154
155     setup_poll_manager(server->poll_manager, size);
156     handle_game_tick(server);
157     if (server->poll_manager->needs_rebuild)
158         fill_poll_array(server, server->poll_manager);
159     if (poll(server->poll_manager->fds, size, 10) > 0)
160         poll_client(server, server->poll_manager);
161 }
```

14.198.1.2 check_client_message()

```
static void check_client_message (
    server_t * server ) [static]
```

Definition at line 59 of file server_run.c.

```
60 {
61     client_t *temp = server->client;
62     client_t *next = NULL;
63
64     if (temp != NULL)
65         temp = temp->next;
66     while (temp != NULL) {
67         next = temp->next;
68         if (temp->client_poll != NULL && temp->client_poll->revents != 0
69             && (temp->client_poll->revents & POLLIN))
70             get_message(server, temp);
71         temp = next;
72     }
73 }
```

14.198.1.3 check_new_connection()

```
static void check_new_connection (
    server_t * server ) [static]
```

Definition at line 53 of file server_run.c.

```
54 {
55     if (server->client->client_poll->revents & POLLIN)
56         new_connection(server);
57 }
```

14.198.1.4 fill_poll_array()

```
static void fill_poll_array (
    server_t * server,
    poll_manager_t * poll mana ) [static]
```

Definition at line 105 of file server_run.c.

```
106 {
107     client_t *current = server->client;
108     int size = server->nfds + 1;
109
110     poll mana->fds[0] = *(current->client_poll);
111     current = current->next;
112     for (int i = 1; i < size && current != NULL; i++) {
113         poll mana->fds[i] = *(current->client_poll);
114         smart_polling(current, poll mana, server, i);
115         current = current->next;
116     }
117     poll mana->needs_rebuild = false;
118 }
```

14.198.1.5 find_client_by_socket()

```
client_t* find_client_by_socket (
```

```

        server_t * server,
        int socket_fd )
Definition at line 17 of file server_run.c.
18 {
19     client_t *temp = server->client;
20
21     if (temp && temp->client_fd == server->s_fd)
22         temp = temp->next;
23     while (temp) {
24         if (temp->client_fd == socket_fd)
25             return temp;
26         temp = temp->next;
27     }
28     printf("Error: Could not find client for socket %d\n", socket_fd);
29     return NULL;
30 }
```

14.198.1.6 handle_game_tick()

```
static void handle_game_tick (
    server_t * server ) [static]
```

Definition at line 136 of file server_run.c.

```

137 {
138     static struct timeval last_tick = {0, 0};
139     struct timeval current_time;
140     long time_diff;
141
142     gettimeofday(&current_time, NULL);
143     time_diff = (current_time.tv_sec - last_tick.tv_sec) * 1000 +
144                 (current_time.tv_usec - last_tick.tv_usec) / 1000;
145     if (time_diff >= (1000 / server->parsed_info->frequence)) {
146         update_game_tick(server);
147         last_tick = current_time;
148     }
149 }
```

14.198.1.7 new_connection()

```
static void new_connection (
    server_t * server ) [static]
```

Definition at line 32 of file server_run.c.

```

33 {
34     struct sockaddr_in client_addr;
35     socklen_t addr_len = sizeof(client_addr);
36     int client_fd;
37     client_t *new_client;
38
39     memset(&client_addr, 0, sizeof(client_addr));
40     client_fd = accept(server->s_fd, (struct sockaddr *)&client_addr,
41                         &addr_len);
42     if (client_fd < 0) {
43         perror("Accept failed");
44         exit(84);
45     }
46     add_fd(server, client_fd);
47     server->nfds += 1;
48     new_client = find_client_by_socket(server, client_fd);
49     if (new_client != NULL)
50         write_command_output(new_client->client_fd, "WELCOME\n");
51 }
```

14.198.1.8 poll_client()

```
static void poll_client (
    server_t * server,
    poll_manager_t * poll mana ) [static]
```

Definition at line 120 of file server_run.c.

```

121 {
122     client_t *current = server->client;
123     int size = server->nfds + 1;
124
125     current->client_poll->revents = poll mana->fds[0].revents;
126     check_new_connection(server);
```

```

127     current = current->next;
128     for (int i = 1; i < size && current != NULL; i++) {
129         current->client_poll->revents = poll_mana->fds[i].revents;
130         current = current->next;
131     }
132     if (server->nfds > 0)
133         check_client_message(server);
134 }
```

14.198.1.9 setup_poll_manager()

```
static void setup_poll_manager (
    poll_manager_t * poll_mana,
    int size ) [static]
```

Definition at line 75 of file server_run.c.

```

76 {
77     if (!poll_mana->fds) {
78         poll_mana->fds = malloc(size * sizeof(struct pollfd));
79         if (!poll_mana->fds)
80             server_err("Poll_mana alloc failed\n");
81         poll_mana->capacity = size;
82     }
83     if (size > poll_mana->capacity) {
84         poll_mana->fds = realloc(poll_mana->fds, size * sizeof(struct pollfd));
85         if (!poll_mana->fds)
86             server_err("Poll_mana realloc failed\n");
87         poll_mana->capacity = size;
88     }
89     poll_mana->needs_rebuild = true;
90 }
```

14.198.1.10 smart_polling()

```
static void smart_polling (
    client_t * current,
    poll_manager_t * poll_mana,
    server_t * server,
    int i ) [static]
```

Definition at line 92 of file server_run.c.

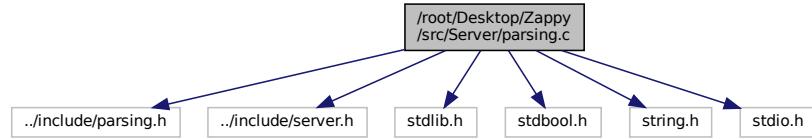
```

94 {
95     if (current->type == GRAPHICAL) {
96         poll_mana->fds[i].events = POLLIN;
97     } else if (current->player &&
98                current->player->busy_until > server->current_tick &&
99                current->player->queue_size >= 10) {
100        poll_mana->fds[i].events = 0;
101    } else
102        poll_mana->fds[i].events = POLLIN;
103 }
```

14.199 /root/Desktop/Zappy/src/Server/parsing.c File Reference

```
#include "../include/parsing.h"
#include "../include/server.h"
#include <stdlib.h>
#include <stdbool.h>
#include <string.h>
#include <stdio.h>
```

Include dependency graph for parsing.c:



Functions

- static bool `check_zero` (char *str)
- static int `parse_port` (char **av, int i, `parsing_info_t` *parsed_info)
- static int `parse_coordinates` (char **av, int i, `parsing_info_t` *parsed_info)

If needed to limit map coordinates add limit to 3rd if.
- static char ** `parse_names` (char **av, int i)
- static int `parse_client_nb` (char **av, int i, `parsing_info_t` *parsed_info)

If needed to limit in any way client_nb change 3rd if.
- static int `parse_frequence` (char **av, int i, `parsing_info_t` *parsed_info)

If needed to limit in any way frequence change 3rd if.
- static bool `is_dup_flag` (char **av, int current_pos, char *flag, `parsing_info_t` *parsed_info)
- static int `check_other_flags` (char **av, int i, `parsing_info_t` *parsed_info)
- static void `check_args` (char **av, int i, `parsing_info_t` *parsed_info)
- void `parse_args` (int ac, char **av, `parsing_info_t` *parsed_info)

14.199.1 Function Documentation

14.199.1.1 `check_args()`

```
static void check_args (
    char ** av,
    int i,
    parsing_info_t * parsed_info ) [static]
```

Definition at line 130 of file parsing.c.

```

131 {
132     if (strcmp(av[i], "-p") == 0 && !is_dup_flag(av, i, "-p", parsed_info)){
133         parsed_info->port = parse_port(av, i, parsed_info);
134         return;
135     }
136     if (strcmp(av[i], "-x") == 0 && !is_dup_flag(av, i, "-x", parsed_info)){
137         parsed_info->width = parse_coordinates(av, i, parsed_info);
138         return;
139     }
140     if (strcmp(av[i], "-y") == 0 && !is_dup_flag(av, i, "-y", parsed_info)){
141         parsed_info->height = parse_coordinates(av, i, parsed_info);
142         return;
143     }
144     if (check_other_flags(av, i, parsed_info) == 1)
145         return;
146     if (av[i][0] == '-'){
147         display_help();
148         exit(84);
149     }
150 }
```

14.199.1.2 check_other_flags()

```
static int check_other_flags (
    char ** av,
    int i,
    parsing_info_t * parsed_info ) [static]
```

Definition at line 113 of file parsing.c.

```
114 {
115     if (strcmp(av[i], "-n") == 0 && !is_dup_flag(av, i, "-n", parsed_info)){
116         parsed_info->names = parse_names(av, i);
117         return 1;
118     }
119     if (strcmp(av[i], "-c") == 0 && !is_dup_flag(av, i, "-c", parsed_info)){
120         parsed_info->client_nb = parse_client_nb(av, i, parsed_info);
121         return 1;
122     }
123     if (strcmp(av[i], "-f") == 0 && !is_dup_flag(av, i, "-f", parsed_info)){
124         parsed_info->frequence = parse_frequence(av, i, parsed_info);
125         return 1;
126     }
127     return 0;
128 }
```

14.199.1.3 check_zero()

```
static bool check_zero (
    char * str ) [static]
```

Definition at line 14 of file parsing.c.

```
15 {
16     if (strcmp(str, "0") == 0)
17         return true;
18     return false;
19 }
```

14.199.1.4 is_dup_flag()

```
static bool is_dup_flag (
    char ** av,
    int current_pos,
    char * flag,
    parsing_info_t * parsed_info ) [static]
```

Definition at line 99 of file parsing.c.

```
101 {
102     char error_msg[256];
103
104     for (int i = 1; i < current_pos; i++) {
105         if (strcmp(av[i], flag) == 0) {
106             sprintf(error_msg, "Duplicate flag: %s", flag);
107             parsing_error(error_msg, parsed_info);
108         }
109     }
110     return false;
111 }
```

14.199.1.5 parse_args()

```
void parse_args (
    int ac,
    char ** av,
    parsing_info_t * parsed_info )
```

Definition at line 152 of file parsing.c.

```
153 {
154     parsed_info->port = 0;
155     parsed_info->width = 0;
156     parsed_info->height = 0;
157     parsed_info->client_nb = 0;
158     parsed_info->frequence = 100;
159     parsed_info->names = NULL;
160     for (int i = 1; i < ac; i++) {
```

```

161         check_args(av, i, parsed_info);
162     }
163 }
```

14.199.1.6 parse_client_nb()

```

static int parse_client_nb (
    char ** av,
    int i,
    parsing_info_t * parsed_info ) [static]
```

If needed to limit in any way client_nb change 3rd if.

Definition at line 70 of file parsing.c.

```

71 {
72     int res = 0;
73
74     if (av[i + 1] == NULL)
75         parsing_error("-c argument null", parsed_info);
76     if (check_zero(av[i + 1]))
77         parsing_error("number of client can't be null", parsed_info);
78     res = atoi(av[i + 1]);
79     if (res < 1)
80         parsing_error("number can't be less than 1", parsed_info);
81     return res;
82 }
```

14.199.1.7 parse_coordinates()

```

static int parse_coordinates (
    char ** av,
    int i,
    parsing_info_t * parsed_info ) [static]
```

If needed to limit map coordinates add limit to 3rd if.

Definition at line 36 of file parsing.c.

```

37 {
38     int res = 0;
39
40     if (av[i + 1] == NULL)
41         parsing_error("-x or -y argument null", parsed_info);
42     if (check_zero(av[i + 1]))
43         parsing_error("coordinates arg can't be null", parsed_info);
44     res = atoi(av[i + 1]);
45     if (res < 1 || res > 100)
46         parsing_error("coordinates can't be less than 1 and more than 100",
47                         parsed_info);
48     return res;
49 }
```

14.199.1.8 parse_frequence()

```

static int parse_frequence (
    char ** av,
    int i,
    parsing_info_t * parsed_info ) [static]
```

If needed to limit in any way frequence change 3rd if.

Definition at line 85 of file parsing.c.

```

86 {
87     int res = 0;
88
89     if (av[i + 1] == NULL)
90         parsing_error("-f argument null", parsed_info);
91     if (check_zero(av[i + 1]))
92         parsing_error("frequence can't be null", parsed_info);
93     res = atoi(av[i + 1]);
94     if (res < 1)
95         parsing_error("frequence can't be less than 1", parsed_info);
96     return res;
97 }
```

14.199.1.9 parse_names()

```
static char** parse_names (
    char ** av,
    int i ) [static]
```

Definition at line 51 of file parsing.c.

```
52 {
53     int j = i + 1;
54     int count = 0;
55     char **result = malloc(sizeof(char *) * 2);
56
57     while (av[j] != NULL && av[j][0] != '-') {
58         result[count] = strdup(av[j]);
59         result = realloc(result, (sizeof(char *) * (count + 2)));
60         if (!result)
61             server_err("Realloc failed in parse_names");
62         j++;
63         count++;
64     }
65     result[count] = NULL;
66     return result;
67 }
```

14.199.1.10 parse_port()

```
static int parse_port (
    char ** av,
    int i,
    parsing_info_t * parsed_info ) [static]
```

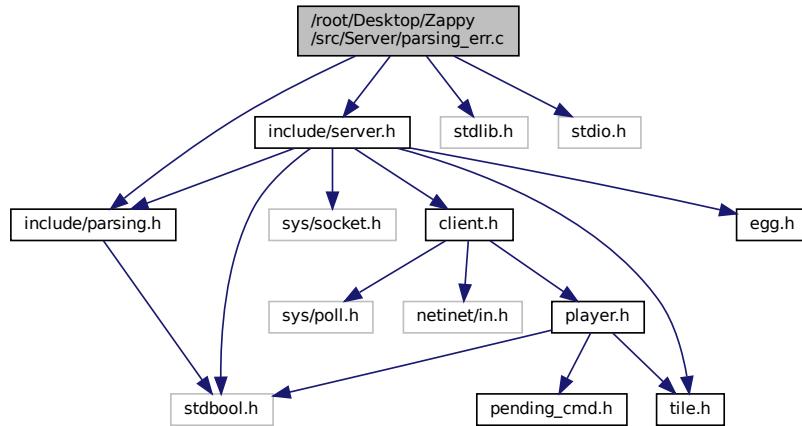
Definition at line 21 of file parsing.c.

```
22 {
23     int res = 0;
24
25     if (av[i + 1] == NULL)
26         parsing_error("-p argument null", parsed_info);
27     if (check_zero(av[i + 1]))
28         return res;
29     res = atoi(av[i + 1]);
30     if (res < 1 || res > 65535)
31         parsing_error("Port needs to be between 1 and 65535", parsed_info);
32     return res;
33 }
```

14.200 /root/Desktop/Zappy/src/Server/parsing_err.c File Reference

```
#include "include/parsing.h"
#include "include/server.h"
#include <stdlib.h>
#include <stdio.h>
```

Include dependency graph for parsing_err.c:



Functions

- static void [free_names](#) (char **names)
- void [parsing_error](#) (char *error_message, [parsing_info_t](#) *parsed_info)

14.200.1 Function Documentation

14.200.1.1 [free_names\(\)](#)

```
static void free_names (
    char ** names ) [static]
```

Definition at line 12 of file parsing_err.c.

```
13 {
14     for (int i = 0; names[i] != NULL; i++) {
15         free(names[i]);
16     }
17     free(names);
18 }
```

14.200.1.2 [parsing_error\(\)](#)

```
void parsing_error (
    char * error_message,
    parsing\_info\_t * parsed_info )
```

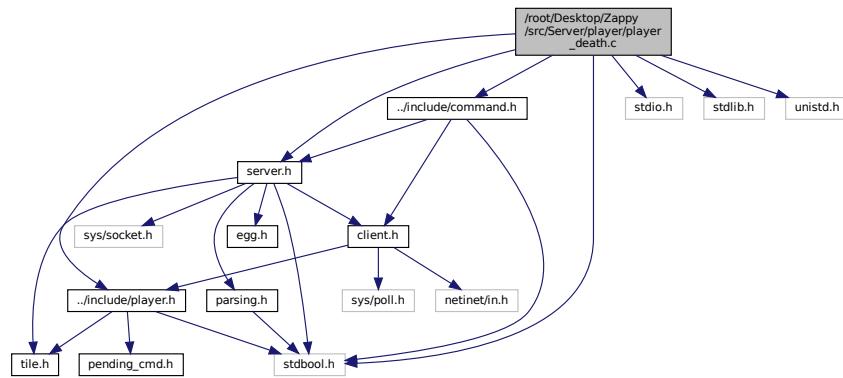
Definition at line 20 of file parsing_err.c.

```
21 {
22     if (parsed_info->names != NULL && parsed_info->names[0] != NULL)
23         free_names(parsed_info->names);
24     printf("%s\n", error_message);
25     display_help();
26     exit(84);
27 }
```

14.201 /root/Desktop/Zappy/src/Server/player/player_death.c File Reference

```
#include "../include/player.h"
#include "../include/command.h"
```

```
#include "../include/server.h"
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <stdbool.h>
Include dependency graph for player_death.c:
```



Functions

- void handle_player_death (server_t *server, client_t *client)
- bool check_player_starvation (server_t *server, client_t *client)

14.201.1 Function Documentation

14.201.1.1 check_player_starvation()

```
bool check_player_starvation (
    server_t * server,
    client_t * client )
```

Definition at line 39 of file player_death.c.

```
40 {
41     int food_amount = 0;
42     bool dead = false;
43
44     if (!client || !client->player || !client->is_fully_connected)
45         return dead;
46     client->player->life--;
47     if (client->player->life <= 0) {
48         food_amount = how_many_in_inventory(client->player, FOOD);
49         if (food_amount > 0) {
50             remove_item_from_inventory(client->player, FOOD, 1);
51             client->player->life = MAX_LIFE_AFTER_FOOD;
52         } else {
53             handle_player_death(server, client);
54             dead = true;
55         }
56     }
57     return dead;
58 }
```

14.201.1.2 handle_player_death()

```
void handle_player_death (
    server_t * server,
    client_t * client )
```

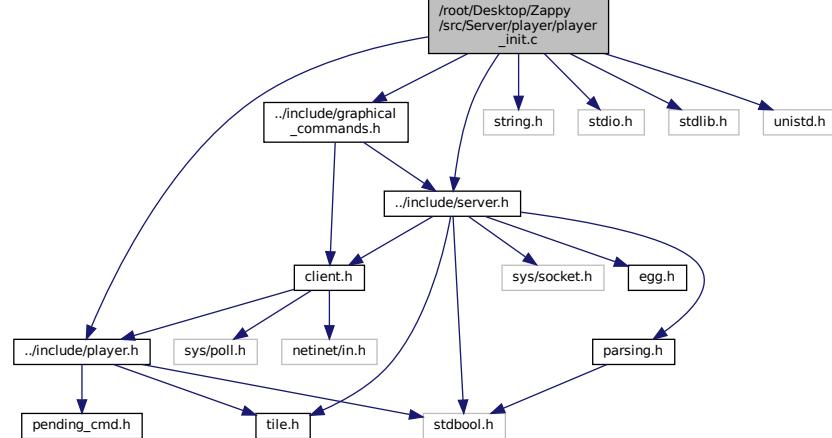
Definition at line 15 of file player_death.c.

```
16 {
17     tile_t *tile;
18
19     if (!client || !client->player)
20         return;
21     write_command_output(client->client_fd, "dead\n");
22     command_pdi(server, client);
23     if (server->map && client->player->pos_y < server->parsed_info->height &&
24         client->player->pos_x < server->parsed_info->width) {
25         tile = &server->map[client->player->pos_y][client->player->pos_x];
26         tile_remove_player(tile, client->client_id);
27     }
28     if (client->player) {
29         free_inventory(client->player);
30         cleanup_player_queue(client->player);
31         cleanup_pending(client->player);
32         free(client->player->team_name);
33         free(client->player);
34         client->player = NULL;
35     }
36     remove_fd(server, client->client_fd);
37 }
```

14.202 /root/Desktop/Zappy/src/Server/player/player_init.c File Reference

```
#include "../include/player.h"
#include "../include/server.h"
#include "../include/graphical_commands.h"
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
Include dependency graph for player_init.c:
```

Include dependency graph for player_init.c:



Functions

- static `egg_t * find_egg_for_team (server_t *server, char *team_name)`
 - void `init_new_player_pos (server_t *server, client_t *new_client)`

The second update of client position is a fallback maybe need more error handling.

- static void `init_pending` (`player_t` *`player`)
 - static void `init_player_team` (`player_t` *`player`, `char` *`player_team`)
 - void `init_player` (`player_t` *`player`, `char` *`player_team`)

14.202.1 Function Documentation

14.202.1.1 find_egg_for_team()

```
static egg_t* find_egg_for_team (
    server_t * server,
    char * team_name ) [static]
```

Definition at line 16 of file player_init.c.

```
17 {
18     egg_t *current = server->eggs;
19
20     if (team_name == NULL)
21         return NULL;
22     while (current != NULL) {
23         if (current->team_name != NULL &&
24             strcmp(current->team_name, team_name) == 0)
25             return current;
26         current = current->next;
27     }
28     return NULL;
29 }
```

14.202.1.2 init_new_player_pos()

```
void init_new_player_pos (
    server_t * server,
    client_t * new_client )
```

The second update of client position is a fallback maybe need more error handling.

Initialise la position d'un nouveau joueur sur la carte.

Definition at line 33 of file player_init.c.

```
34 {
35     int random_x = rand() % server->parsed_info->width;
36     int random_y = rand() % server->parsed_info->height;
37     egg_t *team_egg;
38
39     if (new_client->player->team_name != NULL) {
40         team_egg = find_egg_for_team(server, new_client->player->team_name);
41         if (team_egg != NULL) {
42             new_client->player->pos_x = team_egg->pos_x;
43             new_client->player->pos_y = team_egg->pos_y;
44             tile_add_player(&server->map[team_egg->pos_y][team_egg->pos_x],
45                             new_client->client_id);
46             send_ebo_command(server, team_egg->egg_id);
47             remove_egg(server, team_egg->egg_id,
48                         &server->map[team_egg->pos_y][team_egg->pos_x]);
49         }
50     }
51     new_client->player->pos_x = random_x;
52     new_client->player->pos_y = random_y;
53     tile_add_player(&server->map[random_y][random_x], new_client->client_id);
54 }
55 }
```

14.202.1.3 init_pending()

```
static void init_pending (
    player_t * player ) [static]
```

Definition at line 57 of file player_init.c.

```
58 {
59     player->pending_cmd = calloc(1, sizeof(pending_cmd_t));
60     player->pending_cmd->args = NULL;
61     player->pending_cmd->func = NULL;
62 }
```

14.202.1.4 init_player()

```
void init_player (
    player_t * player,
```

```
char * player_team )
```

Definition at line 73 of file player_init.c.

```
74 {
75     player->pos_x = 0;
76     player->pos_y = 0;
77     player->rotation = UP;
78     player->busy_until = 0;
79     player->queue_size = 0;
80     player->command_queue = calloc(10, sizeof(char *));
81     if (player->command_queue == NULL)
82         server_err("Malloc failed for command queue");
83     player->level = 1;
84     player->life = 126;
85     if (player_team != NULL) {
86         init_player_team(player, player_team);
87     } else
88         player->team_name = NULL;
89     init_inventory(player);
90     player->is_in_incantation = false;
91     player->incantation_leader_id = -1;
92     init_pending(player);
93 }
```

14.202.1.5 init_player_team()

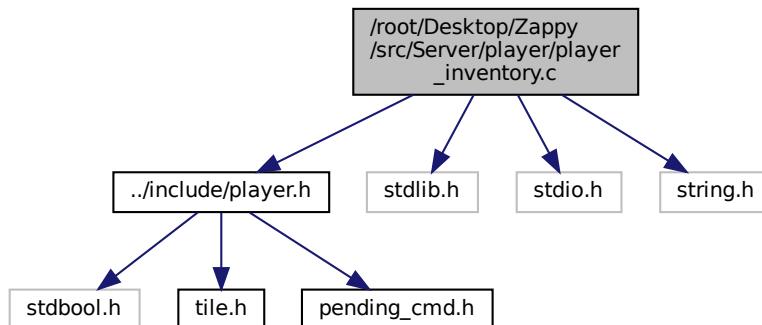
```
static void init_player_team (
    player_t * player,
    char * player_team ) [static]
```

Definition at line 64 of file player_init.c.

```
65 {
66     player->team_name = strdup(player_team);
67     if (player->team_name == NULL) {
68         free(player->command_queue);
69         server_err("Strdup failed for player team name");
70     }
71 }
```

14.203 /root/Desktop/Zappy/src/Server/player/player_inventory.c File Reference

```
#include "../include/player.h"
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
Include dependency graph for player_inventory.c:
```



Functions

- void `init_inventory (player_t *player)`
- void `free_inventory (player_t *player)`
- bool `add_item_to_inventory (player_t *player, resource_type_t type, int q)`
- bool `remove_item_from_inventory (player_t *player, resource_type_t type, int nb)`

14.203.1 Function Documentation

14.203.1.1 add_item_to_inventory()

```
bool add_item_to_inventory (
    player_t * player,
    resource_type_t type,
    int q )
```

Definition at line 33 of file player_inventory.c.

```
34 {
35     if (!player || q < 0 || type >= COUNT)
36         return false;
37     player->inventory[type] += q;
38     return true;
39 }
```

14.203.1.2 free_inventory()

```
void free_inventory (
    player_t * player )
```

Definition at line 23 of file player_inventory.c.

```
24 {
25     if (!player)
26         return;
27     for (int i = 0; i < COUNT; i++) {
28         player->inventory[i] = 0;
29     }
30     player->inventory_size = 0;
31 }
```

14.203.1.3 init_inventory()

```
void init_inventory (
    player_t * player )
```

Definition at line 13 of file player_inventory.c.

```
14 {
15     if (!player)
16         return;
17     for (int i = 0; i < COUNT; i++) {
18         player->inventory[i] = 0;
19     }
20     player->inventory[FOOD] = 10;
21 }
```

14.203.1.4 remove_item_from_inventory()

```
bool remove_item_from_inventory (
    player_t * player,
    resource_type_t type,
    int nb )
```

Definition at line 41 of file player_inventory.c.

```
42 {
43     if (!player || nb <= 0 || type >= COUNT)
44         return false;
45     if (player->inventory[type] == 0) {
46         printf("Item not found in inventory.\n");
```

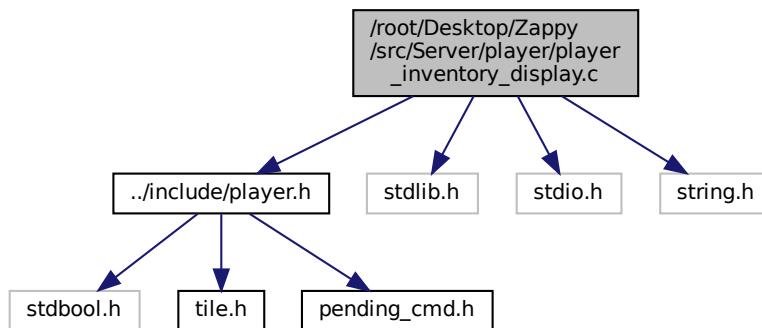
```

47         return false;
48     }
49     if (player->inventory[type] < nb)
50         printf("Warning: Not enough quantity to remove, remove all\n");
51     player->inventory[type] -= nb;
52     if (player->inventory[type] < 0)
53         player->inventory[type] = 0;
54     return true;
55 }

```

14.204 /root/Desktop/Zappy/src/Server/player/player_inventory_display.c File Reference

```
#include "../include/player.h"
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
Include dependency graph for player_inventory_display.c:
```



Functions

- static char * [get_one_item_content](#) (resource_type_t type, int quantity)
- static bool [append_item_string](#) (char **content, size_t *content_length, const char *item_str)
- static void [add_separator](#) (char **content, bool is_last)
- static void [process_one_item](#) (resource_type_t type, int quantity, char **content, size_t *content_length)
- char * [get_inventory_content](#) (player_t *player)

14.204.1 Function Documentation

14.204.1.1 add_separator()

```
static void add_separator (
    char ** content,
    bool is_last ) [static]
```

Definition at line 43 of file player_inventory_display.c.

```

44 {
45     if (!is_last)
46         strcat(*content, ", ");
47     else
48         strcat(*content, "]\n");
49 }
```

14.204.1.2 append_item_string()

```
static bool append_item_string (
    char ** content,
    size_t * content_length,
    const char * item_str ) [static]
```

Definition at line 28 of file player_inventory_display.c.

```
30 {
31     size_t item_length = strlen(item_str);
32     char *temp;
33
34     *content_length += item_length + 3;
35     temp = realloc(*content, *content_length);
36     if (!temp)
37         return false;
38     *content = temp;
39     strcat(*content, item_str);
40     return true;
41 }
```

14.204.1.3 get_inventory_content()

```
char* get_inventory_content (
    player_t * player )
```

Definition at line 67 of file player_inventory_display.c.

```
68 {
69     char *content = strdup("[");
70     size_t content_length = 1;
71     int quantity;
72
73     if (!player)
74         return strdup("[]");
75     for (int i = 0; i < COUNT; i++) {
76         quantity = 0;
77         if (player->inventory[i])
78             quantity = player->inventory[i];
79         process_one_item(i, quantity, &content, &content_length);
80         if (!content)
81             return NULL;
82         add_separator(&content, i == COUNT - 1);
83     }
84     return content;
85 }
```

14.204.1.4 get_one_item_content()

```
static char* get_one_item_content (
    resource_type_t type,
    int quantity ) [static]
```

Definition at line 13 of file player_inventory_display.c.

```
14 {
15     char *name = get_resource_name(type);
16     int total_length = strlen(name) + snprintf(NULL, 0, "%d", quantity) + 3;
17     char *content = malloc(total_length);
18
19     if (!content) {
20         free(name);
21         return NULL;
22     }
23     snprintf(content, total_length, "%s %d", name, quantity);
24     free(name);
25     return content;
26 }
```

14.204.1.5 process_one_item()

```
static void process_one_item (
    resource_type_t type,
    int quantity,
    char ** content,
    size_t * content_length ) [static]
```

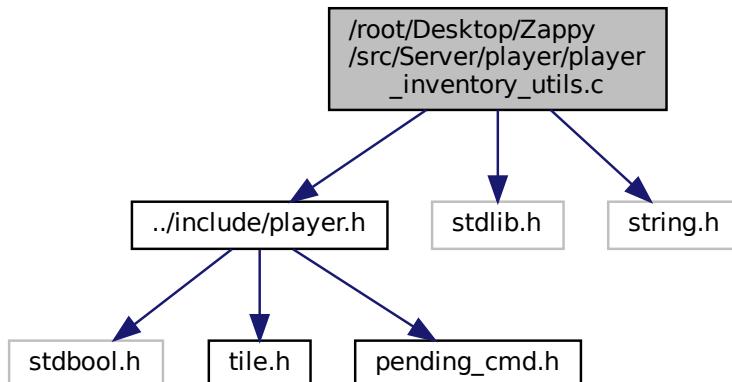
Definition at line 51 of file player_inventory_display.c.

```

53 {
54     char *item_str = get_one_item_content(type, quantity);
55
56     if (!item_str)
57         return;
58     if (!append_item_string(content, content_length, item_str)) {
59         free(item_str);
60         free(*content);
61         *content = NULL;
62         return;
63     }
64     free(item_str);
65 }
```

14.205 /root/Desktop/Zappy/src/Server/player/player_inventory_utils.c File Reference

```
#include "../include/player.h"
#include <stdlib.h>
#include <string.h>
Include dependency graph for player_inventory_utils.c:
```



Functions

- char * [get_resource_name](#) (resource_type_t type)
- bool [inventory_has_item](#) (player_t *player, resource_type_t type)
- int [how_many_in_inventory](#) (player_t *player, resource_type_t type)

14.205.1 Function Documentation

14.205.1.1 [get_resource_name\(\)](#)

```
char* get_resource_name (
    resource_type_t type )
Definition at line 12 of file player_inventory_utils.c.
```

```

13 {
14     switch (type) {
15         case FOOD:
16             return strdup("food");
17         case LINEMATE:
```

```

18         return strdup("linemate");
19     case DERAUMERE:
20         return strdup("deraumere");
21     case SIBUR:
22         return strdup("sibur");
23     case MENDIANE:
24         return strdup("mendiane");
25     case PHIRAS:
26         return strdup("phiras");
27     case THYSTAME:
28         return strdup("thystame");
29     default:
30         return NULL;
31     }
32 }

```

14.205.1.2 how_many_in_inventory()

```
int how_many_in_inventory (
    player_t * player,
    resource_type_t type )
```

Definition at line 41 of file player_inventory_utils.c.

```

42 {
43     if (!player || type >= COUNT)
44         return 0;
45     if (!player->inventory[type])
46         return 0;
47     return player->inventory[type];
48 }
```

14.205.1.3 inventory_has_item()

```
bool inventory_has_item (
    player_t * player,
    resource_type_t type )
```

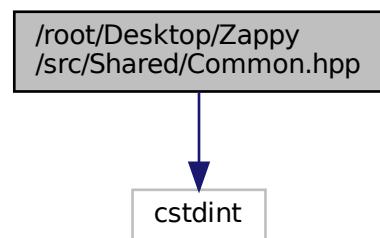
Definition at line 34 of file player_inventory_utils.c.

```

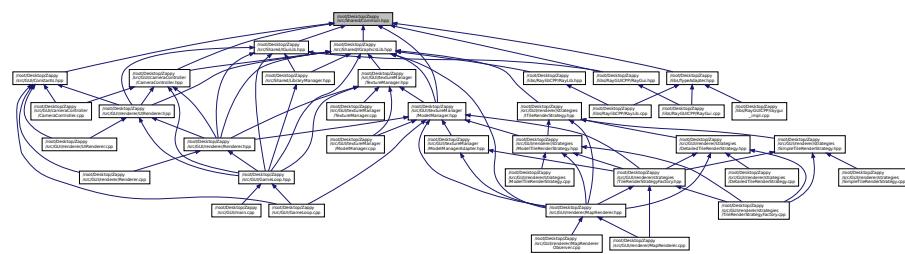
35 {
36     if (!player || type >= COUNT)
37         return false;
38     return player->inventory[type] > 0;
39 }
```

14.206 /root/Desktop/Zappy/src/Shared/Common.hpp File Reference

```
#include <cstdint>
Include dependency graph for Common.hpp:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ZappyTypes::Vector2](#)
- struct [ZappyTypes::Vector3](#)
- struct [ZappyTypes::Color](#)
- struct [ZappyTypes::Rectangle](#)

Namespaces

- [ZappyTypes](#)
- [ZappyTypes::Colors](#)

Enumerations

- enum [ZappyTypes::KeyboardKey](#) {
 [ZappyTypes::Z_KEY_NULL](#) = 0 , [ZappyTypes::Z_KEY_APOSTROPHE](#) = 39 , [ZappyTypes::Z_KEY_COMMA](#) = 44 , [ZappyTypes::Z_KEY_MINUS](#) = 45 ,
 [ZappyTypes::Z_KEY_PERIOD](#) = 46 , [ZappyTypes::Z_KEY_SLASH](#) = 47 , [ZappyTypes::Z_KEY_ZERO](#) = 48 ,
 [ZappyTypes::Z_KEY_ONE](#) = 49 ,
 [ZappyTypes::Z_KEY_TWO](#) = 50 , [ZappyTypes::Z_KEY_THREE](#) = 51 , [ZappyTypes::Z_KEY_FOUR](#) = 52 ,
 [ZappyTypes::Z_KEY_FIVE](#) = 53 ,
 [ZappyTypes::Z_KEY_SIX](#) = 54 , [ZappyTypes::Z_KEY_SEVEN](#) = 55 , [ZappyTypes::Z_KEY_EIGHT](#) = 56 ,
 [ZappyTypes::Z_KEY_NINE](#) = 57 ,
 [ZappyTypes::Z_KEY_SEMICOLON](#) = 59 , [ZappyTypes::Z_KEY_EQUAL](#) = 61 , [ZappyTypes::Z_KEY_A](#) = 65 ,
 [ZappyTypes::Z_KEY_B](#) = 66 ,
 [ZappyTypes::Z_KEY_C](#) = 67 , [ZappyTypes::Z_KEY_D](#) = 68 , [ZappyTypes::Z_KEY_E](#) = 69 , [ZappyTypes::Z_KEY_F](#) = 70 ,
 [ZappyTypes::Z_KEY_G](#) = 71 , [ZappyTypes::Z_KEY_H](#) = 72 , [ZappyTypes::Z_KEY_I](#) = 73 , [ZappyTypes::Z_KEY_J](#) = 74 ,
 [ZappyTypes::Z_KEY_K](#) = 75 , [ZappyTypes::Z_KEY_L](#) = 76 , [ZappyTypes::Z_KEY_M](#) = 77 , [ZappyTypes::Z_KEY_N](#) = 78 ,
 [ZappyTypes::Z_KEY_O](#) = 79 , [ZappyTypes::Z_KEY_P](#) = 80 , [ZappyTypes::Z_KEY_Q](#) = 81 , [ZappyTypes::Z_KEY_R](#) = 82 ,
 [ZappyTypes::Z_KEY_S](#) = 83 , [ZappyTypes::Z_KEY_T](#) = 84 , [ZappyTypes::Z_KEY_U](#) = 85 , [ZappyTypes::Z_KEY_V](#) = 86 ,
 [ZappyTypes::Z_KEY_W](#) = 87 , [ZappyTypes::Z_KEY_X](#) = 88 , [ZappyTypes::Z_KEY_Y](#) = 89 , [ZappyTypes::Z_KEY_Z](#) = 90 ,
 [ZappyTypes::Z_KEY_ESCAPE](#) = 256 , [ZappyTypes::Z_KEY_ENTER](#) = 257 , [ZappyTypes::Z_KEY_SPACE](#) = 32
 }

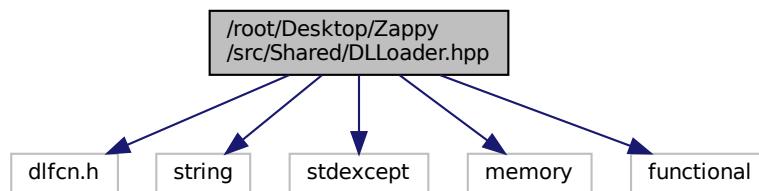
Variables

- static constexpr [ZappyTypes::Color ZappyTypes::Colors::Z_LIGHTGRAY](#) = { 200, 200, 200, 255 }
- static constexpr [ZappyTypes::Color ZappyTypes::Colors::Z_GRAY](#) = { 130, 130, 130, 255 }
- static constexpr [ZappyTypes::Color ZappyTypes::Colors::Z_DARKGRAY](#) = { 80, 80, 80, 255 }

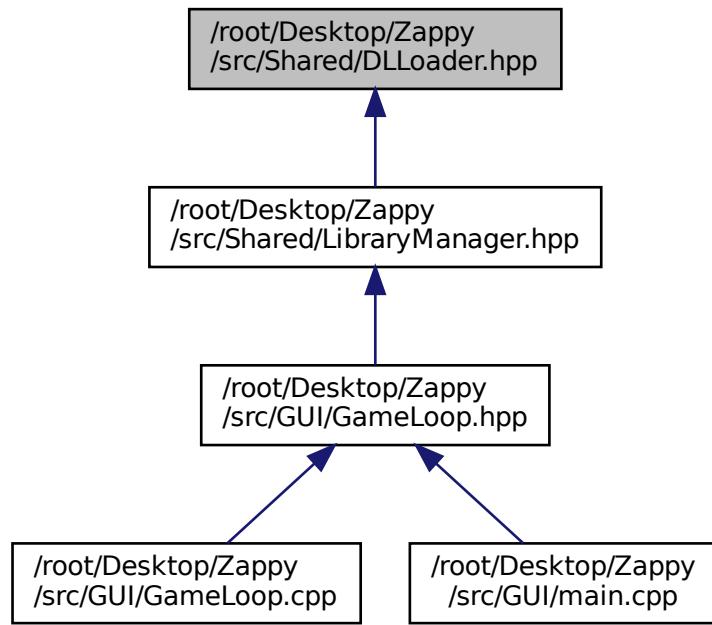
- static constexpr ZappyTypes::Color ZappyTypes::Colors::Z_YELLOW = { 253, 249, 0, 255 }
- static constexpr ZappyTypes::Color ZappyTypes::Colors::Z_GOLD = { 255, 203, 0, 255 }
- static constexpr ZappyTypes::Color ZappyTypes::Colors::Z_ORANGE = { 255, 161, 0, 255 }
- static constexpr ZappyTypes::Color ZappyTypes::Colors::Z_PINK = { 255, 109, 194, 255 }
- static constexpr ZappyTypes::Color ZappyTypes::Colors::Z_RED = { 230, 41, 55, 255 }
- static constexpr ZappyTypes::Color ZappyTypes::Colors::Z_MAROON = { 190, 33, 55, 255 }
- static constexpr ZappyTypes::Color ZappyTypes::Colors::Z_GREEN = { 0, 228, 48, 255 }
- static constexpr ZappyTypes::Color ZappyTypes::Colors::Z_LIME = { 0, 158, 47, 255 }
- static constexpr ZappyTypes::Color ZappyTypes::Colors::Z_DARKGREEN = { 0, 117, 44, 255 }
- static constexpr ZappyTypes::Color ZappyTypes::Colors::Z_SKYBLUE = { 102, 191, 255, 255 }
- static constexpr ZappyTypes::Color ZappyTypes::Colors::Z_BLUE = { 0, 121, 241, 255 }
- static constexpr ZappyTypes::Color ZappyTypes::Colors::Z_DARKBLUE = { 0, 82, 172, 255 }
- static constexpr ZappyTypes::Color ZappyTypes::Colors::Z_PURPLE = { 200, 122, 255, 255 }
- static constexpr ZappyTypes::Color ZappyTypes::Colors::Z_VIOLET = { 135, 60, 190, 255 }
- static constexpr ZappyTypes::Color ZappyTypes::Colors::Z_DARKPURPLE = { 112, 31, 126, 255 }
- static constexpr ZappyTypes::Color ZappyTypes::Colors::Z_BEIGE = { 211, 176, 131, 255 }
- static constexpr ZappyTypes::Color ZappyTypes::Colors::Z_BROWN = { 127, 106, 79, 255 }
- static constexpr ZappyTypes::Color ZappyTypes::Colors::Z_DARKBROWN = { 76, 63, 47, 255 }
- static constexpr ZappyTypes::Color ZappyTypes::Colors::Z_WHITE = { 255, 255, 255, 255 }
- static constexpr ZappyTypes::Color ZappyTypes::Colors::Z_BLACK = { 0, 0, 0, 255 }
- static constexpr ZappyTypes::Color ZappyTypes::Colors::Z_BLANK = { 0, 0, 0, 0 }
- static constexpr ZappyTypes::Color ZappyTypes::Colors::Z_MAGENTA = { 255, 0, 255, 255 }
- static constexpr ZappyTypes::Color ZappyTypes::Colors::Z_RAYWHITE = { 245, 245, 245, 255 }

14.207 /root/Desktop/Zappy/src/Shared/DLLoader.hpp File Reference

```
#include <dlfcn.h>
#include <string>
#include <stdexcept>
#include <memory>
#include <functional>
Include dependency graph for DLLoader.hpp:
```



This graph shows which files directly or indirectly include this file:



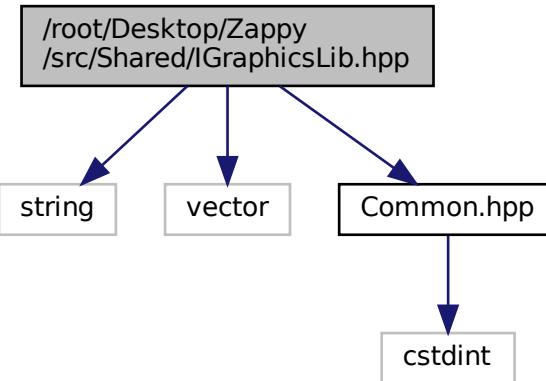
Data Structures

- class [DLLoader< T >](#)

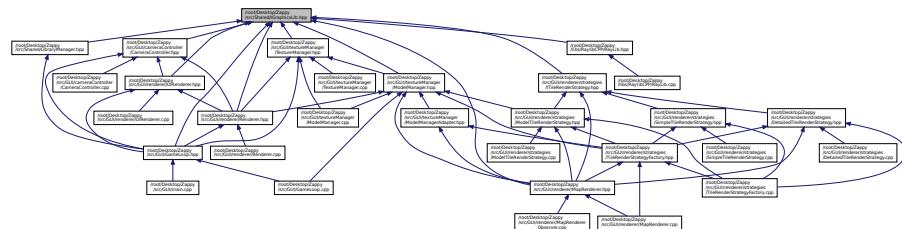
14.208 /root/Desktop/Zappy/src/Shared/IGraphicsLib.hpp File Reference

```
#include <string>
#include <vector>
#include "Common.hpp"
```

Include dependency graph for IGraphicsLib.hpp:



This graph shows which files directly or indirectly include this file:



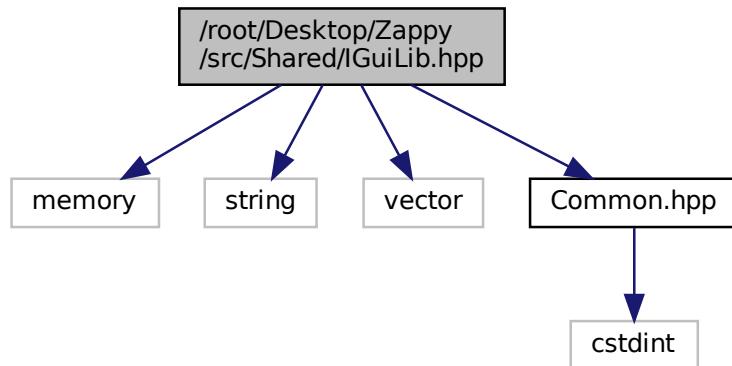
Data Structures

- class [IGraphicsLib](#)

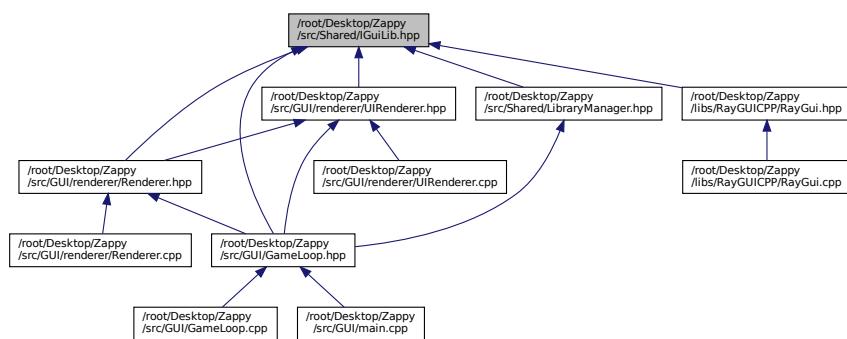
14.209 /root/Desktop/Zappy/src/Shared/IGuiLib.hpp File Reference

```
#include <memory>
#include <string>
#include <vector>
#include "Common.hpp"
```

Include dependency graph for IGuiLib.hpp:



This graph shows which files directly or indirectly include this file:



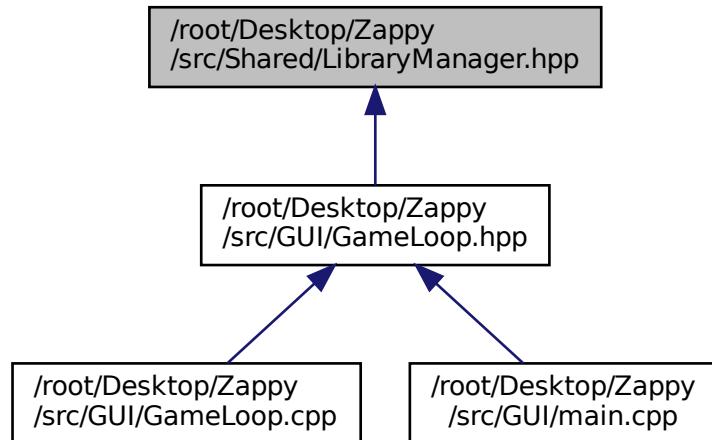
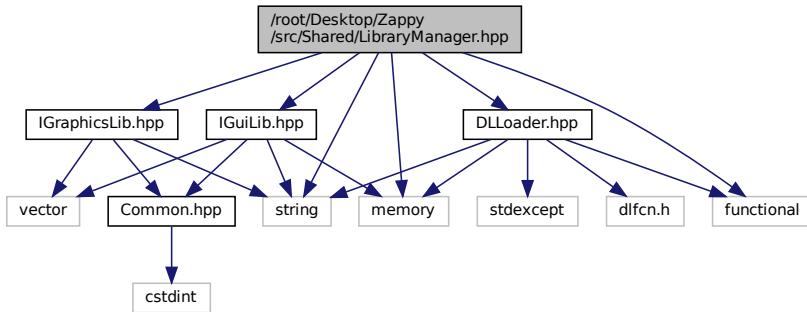
Data Structures

- class [IGuiLib](#)

14.210 /root/Desktop/Zappy/src/Shared/LibraryManager.hpp File Reference

```
#include <string>
#include <memory>
#include <functional>
#include "DLLoader.hpp"
#include "IGraphicsLib.hpp"
#include "IGuiLib.hpp"
```

Include dependency graph for LibraryManager.hpp:



Data Structures

- class [LibraryManager](#)

Index

/root/Desktop/Zappy/README.md, 395
/root/Desktop/Zappy/docs/index.md, 315
/root/Desktop/Zappy/docs/protocol_server_ai.md, 315
/root/Desktop/Zappy/libs/RayGUICPP/README.md, 395
/root/Desktop/Zappy/libs/RayGUICPP/RayGui.cpp, 317
/root/Desktop/Zappy/libs/RayGUICPP/RayGui.hpp, 317
/root/Desktop/Zappy/libs/RayGUICPP/include/RayGuiEncoder.h, 315
/root/Desktop/Zappy/libs/RayGUICPP/raygui_impl.cpp, 318
/root/Desktop/Zappy/libs/RaylibCPP/README.md, 395
/root/Desktop/Zappy/libs/RaylibCPP/RayLib.cpp, 342
/root/Desktop/Zappy/libs/RaylibCPP/RayLib.hpp, 343
/root/Desktop/Zappy/libs/RaylibCPP/audio/Audio.cpp, 320
/root/Desktop/Zappy/libs/RaylibCPP/audio/Audio.hpp, 320
/root/Desktop/Zappy/libs/RaylibCPP/camera/Camera.cpp, 322
/root/Desktop/Zappy/libs/RaylibCPP/camera/Camera.hpp, 322
/root/Desktop/Zappy/libs/RaylibCPP/font/Font.cpp, 323
/root/Desktop/Zappy/libs/RaylibCPP/font/Font.hpp, 324
/root/Desktop/Zappy/libs/RaylibCPP/font/Text3D.cpp, 325
/root/Desktop/Zappy/libs/RaylibCPP/font/Text3D.hpp, 326
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DCode.h, 327
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DConfig.cpp, 328
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DConfig.hpp, 329
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DCore.hpp, 330
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DHelper.cpp, 332
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DHelper.hpp, 333
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DMeasure.h, 334
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DMeasure.hpp, 335
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DParse.h, 335
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DParse.hpp, 335
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DParser.h, 336
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DParser.hpp, 336
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DRenderer.cpp, 337
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DRenderer.hpp, 338
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DWave.cpp, 339
/root/Desktop/Zappy/libs/RaylibCPP/font/text3d/Text3DWave.hpp, 339
/root/Desktop/Zappy/libs/RaylibCPP/input/Input.cpp, 340
/root/Desktop/Zappy/libs/RaylibCPP/input/Input.hpp, 341
/root/Desktop/Zappy/libs/RaylibCPP/shape/Shape.cpp, 344
/root/Desktop/Zappy/libs/RaylibCPP/shape/Shape.hpp, 344
/root/Desktop/Zappy/libs/RaylibCPP/texture/Texture.cpp, 345
/root/Desktop/Zappy/libs/RaylibCPP/texture/Texture.hpp, 346
/root/Desktop/Zappy/libs/RaylibCPP/utils/Utils.cpp, 347
/root/Desktop/Zappy/libs/RaylibCPP/utils/Utils.hpp, 348
/root/Desktop/Zappy/libs/RaylibCPP/window/Window.cpp, 349
/root/Desktop/Zappy/libs/RaylibCPP/window/Window.hpp, 350
/root/Desktop/Zappy/libs>TypeAdapter.hpp, 351
/root/Desktop/Zappy/src/AI/agent/actions.py, 352
/root/Desktop/Zappy/src/AI/agent/agent.py, 352
/root/Desktop/Zappy/src/AI/agent/behaviors.py, 352
/root/Desktop/Zappy/src/AI/agent/broadcastManager.py, 352
/root/Desktop/Zappy/src/AI/agent/decisionManager.py, 352
/root/Desktop/Zappy/src/AI/agent/socketManager.py, 353
/root/Desktop/Zappy/src/AI/constants/resources.py, 353
/root/Desktop/Zappy/src/AI/constants/roles.py, 353
/root/Desktop/Zappy/src/AI/constants/upgrades.py, 353
/root/Desktop/Zappy/src/AI/logger/logger.py, 353
/root/Desktop/Zappy/src/AI/main.py, 353
/root/Desktop/Zappy/src/AI/parser/paringArgsClass.py, 354
/root/Desktop/Zappy/src/AI/requirements.txt, 354
/root/Desktop/Zappy/src/AI/utils/encryption.py, 354
/root/Desktop/Zappy/src/AI/utils/input.py, 354
/root/Desktop/Zappy/src/AI/utils/zappy.py, 355
/root/Desktop/Zappy/src/GUI/Constants.hpp, 356

/root/Desktop/Zappy/src/GUI/GameLoop.cpp, 360	/root/Desktop/Zappy/src/GUI/network/protocol/messageData/PlayerInventory.hpp, 383
/root/Desktop/Zappy/src/GUI/GameLoop.hpp, 360	/root/Desktop/Zappy/src/GUI/network/protocol/messageData/ResourceData.hpp, 383
/root/Desktop/Zappy/src/GUI/README.md, 395	/root/Desktop/Zappy/src/GUI/network/protocol/messageData/TeamNameData.hpp, 384
/root/Desktop/Zappy/src/GUI/cameraController/CameraController.cpp, 355	/root/Desktop/Zappy/src/GUI/network/protocol/messageData/ServerMessageData.hpp, 384
/root/Desktop/Zappy/src/GUI/cameraController/CameraController.hpp, 355	/root/Desktop/Zappy/src/GUI/network/protocol/messageData/TeamNameData.hpp, 385
/root/Desktop/Zappy/src/GUI/graphicalContext/GraphicalContext.cpp, 361	/root/Desktop/Zappy/src/GUI/network/protocol/messageData/TileContentData.hpp, 386
/root/Desktop/Zappy/src/GUI/graphicalContext/GraphicalContext.hpp, 361	/root/Desktop/Zappy/src/GUI/network/protocol/messageData/TimeUnitData.hpp, 386
/root/Desktop/Zappy/src/GUI/main.cpp, 363	/root/Desktop/Zappy/src/GUI/network/utils/SystemWrapper.cpp, 388
/root/Desktop/Zappy/src/GUI/network/buffer/MessageQueue.cpp, 364	/root/Desktop/Zappy/src/GUI/network/utils/SystemWrapper.hpp, 390
/root/Desktop/Zappy/src/GUI/network/buffer/MessageQueue.hpp, 364	/root/Desktop/Zappy/src/GUI/network/utils/SystemWrapper.hpp, 391
/root/Desktop/Zappy/src/GUI/network/connection/TcpConnection.cpp, 365	/root/Desktop/Zappy/src/GUI/parsing/ParsingCLI.cpp, 393
/root/Desktop/Zappy/src/GUI/network/connection/TcpConnection.hpp, 365	/root/Desktop/Zappy/src/GUI/parsing/ParsingCLI.hpp, 393
/root/Desktop/Zappy/src/GUI/network/networkManager/NetworkManager.cpp, 367	/root/Desktop/Zappy/src/GUI/renderer/MapRenderer.cpp, 395
/root/Desktop/Zappy/src/GUI/network/networkManager/NetworkManager.hpp, 367	/root/Desktop/Zappy/src/GUI/renderer/MapRenderer.hpp, 395
/root/Desktop/Zappy/src/GUI/network/networkManager/NetworkManager.cpp, 368	/root/Desktop/Zappy/src/GUI/renderer/MapRendererObserver.cpp, 396
/root/Desktop/Zappy/src/GUI/network/networkThread/NetworkThread.cpp, 369	/root/Desktop/Zappy/src/GUI/renderer/MapRendererObserver.hpp, 396
/root/Desktop/Zappy/src/GUI/network/networkThread/NetworkThread.hpp, 369	/root/Desktop/Zappy/src/GUI/renderer/MapRendererObserver.hpp, 397
/root/Desktop/Zappy/src/GUI/network/protocol/HeaderMessage.h, 370	/root/Desktop/Zappy/src/GUI/renderer/Renderer.cpp, 397
/root/Desktop/Zappy/src/GUI/network/protocol/HeaderMessage.hpp, 370	/root/Desktop/Zappy/src/GUI/renderer/Renderer.hpp, 398
/root/Desktop/Zappy/src/GUI/network/protocol/Message.cpp, 373	/root/Desktop/Zappy/src/GUI/renderer/UIRenderer.cpp, 405
/root/Desktop/Zappy/src/GUI/network/protocol/Message.hpp, 373	/root/Desktop/Zappy/src/GUI/renderer/UIRenderer.hpp, 406
/root/Desktop/Zappy/src/GUI/network/protocol/ProtocolParser.cpp, 389	/root/Desktop/Zappy/src/GUI/renderer/strategies/DetailedTileRenderStrategy.hpp, 399
/root/Desktop/Zappy/src/GUI/network/protocol/ProtocolParser.hpp, 389	/root/Desktop/Zappy/src/GUI/renderer/strategies/DetailedTileRenderStrategy.hpp, 399
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/PlayerInventory.hpp, 374	/root/Desktop/Zappy/src/GUI/renderer/strategies/ITileRenderStrategy.hpp, 400
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/ResourceData.hpp, 375	/root/Desktop/Zappy/src/GUI/renderer/strategies/ModelTileRenderStrategy.hpp, 401
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/TeamNameData.hpp, 376	/root/Desktop/Zappy/src/GUI/renderer/strategies/ModelTileRenderStrategy.hpp, 401
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/ServerMessageData.hpp, 377	/root/Desktop/Zappy/src/GUI/renderer/strategies/SimpleTileRenderStrategy.hpp, 402
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/TeamNameData.hpp, 378	/root/Desktop/Zappy/src/GUI/renderer/strategies/SimpleTileRenderStrategy.hpp, 403
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/TimeUnitData.hpp, 379	/root/Desktop/Zappy/src/GUI/renderer/strategies/TileRenderStrategyFactor.hpp, 404
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/TimeUnitData.hpp, 379	/root/Desktop/Zappy/src/GUI/renderer/strategies/TileRenderStrategyFactor.hpp, 404
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/TileContentData.hpp, 380	/root/Desktop/Zappy/src/GUI/renderer/strategies/TileRenderStrategyFactor.hpp, 404
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/TileContentData.hpp, 381	/root/Desktop/Zappy/src/GUI/shared/exception/AException.cpp, 407
/root/Desktop/Zappy/src/GUI/network/protocol/messageData/TileContentData.hpp, 382	/root/Desktop/Zappy/src/GUI/shared/exception/AException.hpp, 408

/root/Desktop/Zappy/src/GUI/shared/exception/IException.hpp,	466	
408		/root/Desktop/Zappy/src/Server/graphical_command/command_pbc.c,
/root/Desktop/Zappy/src/GUI/textureManager/ModelManager.cpp,	467	409
410		/root/Desktop/Zappy/src/Server/graphical_command/command_pdi.c,
/root/Desktop/Zappy/src/GUI/textureManager/ModelManager.hpp,	468	411
410		/root/Desktop/Zappy/src/Server/graphical_command/command_pdr.c,
/root/Desktop/Zappy/src/GUI/textureManager/ModelManagerAdapter.hpp,	469	411
413		/root/Desktop/Zappy/src/Server/graphical_command/command_pex.c,
/root/Desktop/Zappy/src/GUI/textureManager/TextureManager.cpp,	470	412
412		/root/Desktop/Zappy/src/Server/graphical_command/command_pkf.c,
/root/Desktop/Zappy/src/GUI/textureManager/TextureManager.hpp,	471	413
413		/root/Desktop/Zappy/src/Server/graphical_command/command_pgt.c,
/root/Desktop/Zappy/src/Server/README.md,	395	472
/root/Desktop/Zappy/src/Server/buffer/circular_buffer.c,	414	/root/Desktop/Zappy/src/Server/graphical_command/command_pic.c,
414		473
/root/Desktop/Zappy/src/Server/command/broadcast.c,	416	/root/Desktop/Zappy/src/Server/graphical_command/command_pie.c,
416		474
/root/Desktop/Zappy/src/Server/command/connect_nbr.c,	419	/root/Desktop/Zappy/src/Server/graphical_command/command_pin.c,
419		476
/root/Desktop/Zappy/src/Server/command/eject.c,	421	/root/Desktop/Zappy/src/Server/graphical_command/command_plv.c,
/root/Desktop/Zappy/src/Server/command/finish_incantation.c,	425	478
425		/root/Desktop/Zappy/src/Server/graphical_command/command_pnw.c,
/root/Desktop/Zappy/src/Server/command/fork.c,	427	480
/root/Desktop/Zappy/src/Server/command/forward.c,	429	/root/Desktop/Zappy/src/Server/graphical_command/command_ppo.c,
429		482
/root/Desktop/Zappy/src/Server/command/inventory.c,	430	/root/Desktop/Zappy/src/Server/graphical_command/command_seg.c,
430		483
/root/Desktop/Zappy/src/Server/command/left.c,	431	/root/Desktop/Zappy/src/Server/graphical_command/command_sgt.c,
/root/Desktop/Zappy/src/Server/command/look.c,	433	485
/root/Desktop/Zappy/src/Server/command/look_utils.c,	435	/root/Desktop/Zappy/src/Server/graphical_command/command_smg.c,
435		486
/root/Desktop/Zappy/src/Server/command/parse_command	438	/root/Desktop/Zappy/src/Server/graphical_command/command_sst.c,
		487
/root/Desktop/Zappy/src/Server/command/parse_command	441	/root/Desktop/Zappy/src/Server/graphical_command/command_tna.c,
		488
/root/Desktop/Zappy/src/Server/command/parse_command	443	/root/Desktop/Zappy/src/Server/help.c,
		489
/root/Desktop/Zappy/src/Server/command/pending_cmd_utils.c,	445	/root/Desktop/Zappy/src/Server/include/circular_buffer.h,
445		
/root/Desktop/Zappy/src/Server/command/right.c,	446	/root/Desktop/Zappy/src/Server/include/client.h,
/root/Desktop/Zappy/src/Server/command/set_object.c,	447	493
447		/root/Desktop/Zappy/src/Server/include/command.h,
/root/Desktop/Zappy/src/Server/command/start_incantation	448	495
		/root/Desktop/Zappy/src/Server/include/egg.h,
		514
/root/Desktop/Zappy/src/Server/command/take_object.c,	451	/root/Desktop/Zappy/src/Server/include/graphical_commands.h,
451		515
/root/Desktop/Zappy/src/Server/command/tile_to_str.c,	453	/root/Desktop/Zappy/src/Server/include/parsing.h,
453		522
/root/Desktop/Zappy/src/Server/free.c,	455	/root/Desktop/Zappy/src/Server/include/pending_cmd.h,
		524
/root/Desktop/Zappy/src/Server/graphical_command/command	458	/root/Desktop/Zappy/src/Server/include/pending_cmd_utils.h,
		525
/root/Desktop/Zappy/src/Server/graphical_command/command	461	/root/Desktop/Zappy/src/Server/include/player.h,
		527
/root/Desktop/Zappy/src/Server/graphical_command/command	462	/root/Desktop/Zappy/src/Server/include/server.h,
		531
/root/Desktop/Zappy/src/Server/graphical_command/command	462	/root/Desktop/Zappy/src/Server/include.tile.h,
		540
/root/Desktop/Zappy/src/Server/graphical_command/command	462	/root/Desktop/Zappy/src/Server/main.c,
		544
/root/Desktop/Zappy/src/Server/graphical_command/command	462	/root/Desktop/Zappy/src/Server/map/egg.c,
		548
/root/Desktop/Zappy/src/Server/graphical_command/command_enw	464	/root/Desktop/Zappy/src/Server/map/map_creation.c,
		551
/root/Desktop/Zappy/src/Server/graphical_command/command_msz	464	/root/Desktop/Zappy/src/Server/map/resources_map.c,
		552

/root/Desktop/Zappy/src/Server/map/tick_update.c, 554
 /root/Desktop/Zappy/src/Server/map/tile.c, 556
 /root/Desktop/Zappy/src/Server/network/array_function.c, 558
 /root/Desktop/Zappy/src/Server/network/client_node.c, 561
 /root/Desktop/Zappy/src/Server/network/connection.c, 563
 /root/Desktop/Zappy/src/Server/network/connection_err.c, 567
 /root/Desktop/Zappy/src/Server/network/connection_utils.c, 569
 /root/Desktop/Zappy/src/Server/network/graphical_client.c, 570
 /root/Desktop/Zappy/src/Server/network/queue_connection.c, 573
 /root/Desktop/Zappy/src/Server/network/server_run.c, 575
 /root/Desktop/Zappy/src/Server/parsing.c, 578
 /root/Desktop/Zappy/src/Server/parsing_err.c, 582
 /root/Desktop/Zappy/src/Server/player/player_death.c, 583
 /root/Desktop/Zappy/src/Server/player/player_init.c, 585
 /root/Desktop/Zappy/src/Server/player/player_inventory.c, 587
 /root/Desktop/Zappy/src/Server/player/player_inventory_display.c, 589
 /root/Desktop/Zappy/src/Server/player/player_inventory_utils.c, 591
 /root/Desktop/Zappy/src/Shared/Common.hpp, 592
 /root/Desktop/Zappy/src/Shared/DLLoader.hpp, 594
 /root/Desktop/Zappy/src/Shared/IGraphicsLib.hpp, 595
 /root/Desktop/Zappy/src/Shared/IGuiLib.hpp, 596
 /root/Desktop/Zappy/src/Shared/LibraryManager.hpp, 597
`__init__`
 agent.Agent, 70
 behaviors.Behavior, 75
 broadcastManager.BroadcastManager, 78
 decisionManager.DecisionManager, 92
 logger.Logger, 153
 paringArgsClass.parseArgs, 188
 paringArgsClass.ParsingArgsError, 193
 socketManager.SocketManager, 265
`~AException`
 AException, 68
`~Audio`
 raylibcpp::Audio, 74
`~Camera2DWrap`
 raylibcpp::Camera2DWrap, 80
`~Camera3DWrap`
 raylibcpp::Camera3DWrap, 81
`~CameraController`
 CameraController, 83
`~DLLoader`
 DLLoader< T >, 96
`~Font`
 raylibcpp::Font, 106
`~GameLoop`
 GameLoop, 107
`~GraphicalContext`
 GraphicalContext, 115
`~IException`
 IException, 121
`~IGraphicalContextObserver`
 IGraphicalContextObserver, 123
`~IGraphicsLib`
 IGraphicsLib, 125
`~IGuiLib`
 IGuiLib, 133
`~IMessageData`
 IMessageData, 141
`~ITileRenderStrategy`
 Zappy::ITileRenderStrategy, 148
`~Message`
 Message, 164
`~MessageQueue`
 MessageQueue, 167
`~ModelWrap`
 raylibcpp::ModelWrap, 179
`~MusicWrap`
 raylibcpp::MusicWrap, 181
`~NetworkManager`
 NetworkManager, 184
`~NetworkThread`
 NetworkThread, 187
`~ParsingCLI`
 ParsingCLI, 194
`~ProtocolParser`
 ProtocolParser, 210
`~RayGui`
 RayGui, 220
 RayGUICPP::RayGui, 228
`~RayLib`
 RayLib, 230
`~Renderer`
 Renderer, 243
`~SafeBuffer`
 SystemWrapper::SafeBuffer, 250
`~SoundWrap`
 raylibcpp::SoundWrap, 268
`~TcpConnection`
 TcpConnection, 271
`~Text3D`
 raylibcpp::Text3D, 275
`~Texture`
 raylibcpp::Texture, 286
`~Texture3DWrap`
 raylibcpp::Texture3DWrap, 287
`~UIRenderer`
 UIRenderer, 303
`~Window`
 raylibcpp::Window, 312
`a`
 ZappyTypes::Color, 90
 accepted_types

command_data_s, 92
actions, 41
 go_take_item, 41
 go_to_pos_with_distance, 41
 got_to_dir, 42
 take_all_of_item_here, 42
 take_everything_here, 42
add
 raylibcpp::Vector3Wrap, 309
add_egg
 egg.c, 548
 egg.h, 514
add_egg_to_tile
 egg.c, 549
 tile.h, 542
add_fd
 connection.c, 564
 server.h, 533
add_graphic_client
 client.h, 494
 graphical_client.c, 571
add_item_to_inventory
 player.h, 528
 player_inventory.c, 588
add_pending_cmd
 command.h, 497
 parse_command_utils.c, 442
add_resources
 tile_to_str.c, 454
add_separator
 player_inventory_display.c, 589
add_to_circular_buffer
 circular_buffer.c, 415
 circular_buffer.h, 492
add_to_command_queue
 queue_connection.c, 573
 server.h, 533
add_to_graphical_list
 graphical_client.c, 571
add_token
 array_function.c, 558
addIntParam
 Message, 164
addName
 TeamNameData, 274
addObserver
 GraphicalContext, 116
addPlayerId
 IncantationData, 142
addStringParam
 Message, 164
adjustWaveIntensity
 raylibcpp::Text3DConfig, 277
adjustWaveSpeed
 raylibcpp::Text3DConfig, 277
AException, 67
 ~AException, 68
 AException, 68
getFormattedMessage, 68
getMessage, 69
getType, 69
what, 69
agent, 43
 behaviors.Behavior, 76
 broadcastManager.BroadcastManager, 79
 decisionManager.DecisionManager, 93
agent.Agent, 69
 __init__, 70
 broadcastManager, 72
 current_behaviour, 72
 decisionManager, 72
 get_message, 70
 has_messages, 71
 id, 73
 ip, 73
 level, 73
 logger, 73
 map_size_x, 73
 map_size_y, 73
 port, 73
 process_server_message, 71
 run, 71
 send_command, 71
 sock, 73
 socketManager, 73
 start, 72
 stop, 72
 team, 73
AI
 client.h, 494
angleX
 CameraController, 83
angleY
 CameraController, 83
append_item_string
 player_inventory_display.c, 589
applyWaveEffect
 raylibcpp::Text3DWave, 284
args
 main, 45
 paringArgsClass.parseArgs, 191
 pending_cmd_s, 196
arr_len
 array_function.c, 558
 parsing.h, 523
array_function.c
 add_token, 558
 arr_len, 558
 check_if_failed, 559
 count_valid_tokens, 559
 fill_result_array, 559
 free_cpy_arr, 560
 init_result_array, 560
 next_valid_token, 560
 process_tokens, 560
 str_to_word_arr, 561

Audio
 raylibcpp::Audio, 74

b
 ZappyTypes::Color, 90

BCT_HEADER
 HeaderMessage.hpp, 371

BeginCamera3D
 IGraphicsLib, 125
 RayLib, 231

BeginDrawing
 IGraphicsLib, 125
 RayLib, 231

beginDrawing
 raylibcpp::Window, 312

beginMode
 raylibcpp::Camera3DWrap, 82

behaviors, 43
 decisionManager.DecisionManager, 93

behaviors.Behavior, 75
 __init__, 75
 agent, 76
 execute, 76

behaviors.DysonBehavior, 98
 execute, 99

behaviors.GetFoodAndMineralsBehavior, 109
 execute, 110

behaviors.GetFoodBehavior, 111
 execute, 111

behaviors.GetMineralsBehavior, 112
 execute, 113

behaviors.UpgradeBehavior, 305
 execute, 306

bind
 raylibcpp::Texture3DWrap, 288

BindTexture3D
 IGraphicsLib, 125
 RayLib, 231

boundingBoxMax
 Model3D, 169

boundingBoxMin
 Model3D, 169

Broadcast
 IMessageData.hpp, 378

broadcast
 broadcast.c, 416
 command.h, 498

broadcast.c
 broadcast, 416
 build_broadcast_message, 417
 calculate_direction, 417
 calculate_message_length, 417
 calculate_shortest_distance_x, 418
 calculate_shortest_distance_y, 418
 concatenate_message_parts, 418
 get_base_direction, 418
 send_broadcast_to_client, 419

BroadcastData, 76
 BroadcastData, 77

getMessage, 77
 getPlayerId, 77
 getType, 77
 setMessage, 78
 setPlayerId, 78

broadcastManager, 43
 agent.Agent, 72

broadcastManager.BroadcastManager, 78
 __init__, 78
 agent, 79
 manage_broadcast, 78

buffer
 circular_buffer_s, 87
 socketManager.SocketManager, 266

BUFFER_SIZE
 circular_buffer.h, 491

build_broadcast_message
 broadcast.c, 417

busy_until
 player_s, 198

ButtonPressed
 IGuiLib, 133
 RayGui, 220

calcul_size_pnw_command
 command_pnw.c, 480

calculate_direction
 broadcast.c, 417

calculate_message_length
 broadcast.c, 417

calculate_shortest_distance_x
 broadcast.c, 418

calculate_shortest_distance_y
 broadcast.c, 418

calculate_size_bct_command
 command_bct.c, 459

calculate_size_pin_command
 command_pin.c, 476

calculate_total_size
 tile_to_str.c, 454

calculateAdvanceX
 raylibcpp::Text3DHelper, 48

calculateGlyphDimensions
 raylibcpp::Text3DHelper, 49

calculateGlyphPosition
 raylibcpp::Text3DHelper, 49

calculateScale
 raylibcpp::Text3DHelper, 49

calculateSourceRectangle
 raylibcpp::Text3DHelper, 49

calculateTextureCoordinates
 raylibcpp::Text3DHelper, 50

Camera2DWrap
 raylibcpp::Camera2DWrap, 80

Camera3DWrap
 raylibcpp::Camera3DWrap, 81

CameraController, 82
 ~CameraController, 83
 angleX, 83

angleY, 83
CameraController, 83
distance, 83
handleKeyboardInput, 83
handleMouseInput, 84
init, 85
isDragging, 85
position, 85
reset, 85
setDragging, 85
setMapDimensions, 85
setMousePosition, 86
target, 86
up, 86
update, 86
updateCameraPosition, 86
can_connect
 command.h, 498
 parse_command_utils_bis.c, 444
can_start_incantation
 command.h, 499
 start_incantation.c, 449
capacity
 poll_manager_s, 208
change_map_pos
 forward.c, 429
change_rot
 left.c, 432
 right.c, 446
charIndex
 raylibcpp::RenderState, 246
check_args
 parsing.c, 579
check_client
 server.h, 533
 server_run.c, 575
check_client_message
 server_run.c, 576
check_command
 parse_command.c, 438
check_disconnect
 parse_command.c, 439
check_down
 look_utils.c, 436
check_if_failed
 array_function.c, 559
check_left
 look_utils.c, 436
check_mini_args
 main.c, 545
check_new_connection
 server_run.c, 576
check_other_flags
 parsing.c, 579
check_player_starvation
 command.h, 499
 player_death.c, 584
check_rebuild
 queue_connection.c, 574
check_right
 look_utils.c, 437
check_rota_tiles
 command.h, 499
 look_utils.c, 437
check_up
 look_utils.c, 437
check_win
 finish_incantation.c, 425
check_zero
 parsing.c, 580
checkValidityCLI
 ParsingCLI, 195
child_pids
 main, 46
circular_buffer.c
 add_to_circular_buffer, 415
 extract_command, 415
 find_command_end, 415
 init_circular_buffer, 415
circular_buffer.h
 add_to_circular_buffer, 492
 BUFFER_SIZE, 491
 circular_buffer_t, 491
 extract_command, 492
 find_command_end, 492
 init_circular_buffer, 492
circular_buffer_s, 87
 buffer, 87
 count, 87
 end, 88
 start, 88
circular_buffer_t
 circular_buffer.h, 491
cleanup_client
 command.h, 500
 parse_command_utils.c, 442
cleanup_pending
 command.h, 500
 parse_command_utils.c, 443
cleanup_player_client
 client_node.c, 562
cleanup_player_queue
 command.h, 500
 parse_command_utils.c, 443
clear
 MessageQueue, 167
 raylibcpp::Window, 312
ClearBackground
 IGraphicsLib, 125
 RayLib, 231
client
 graphical_client_s, 114
 server_s, 257
client.h
 add_graphic_client, 494
 AI, 494

client_t, 493
 client_type_e, 494
 GRAPHICAL, 494
 graphical_client_t, 494
 remove_graphic_client, 494
 send_map_info_to_one_client, 494
 send_message_to_all_graphic, 495
 client_add
 client_s, 89
 client_fd
 client_s, 89
 client_id
 client_s, 89
 client_nb
 parsing_info_s, 192
 client_node.c
 cleanup_player_client, 562
 free_node, 562
 client_poll
 client_s, 89
 client_s, 88
 client_add, 89
 client_fd, 89
 client_id, 89
 client_poll, 89
 is_fully_connected, 89
 next, 89
 player, 89
 type, 89
 client_t
 client.h, 493
 pending_cmd.h, 525
 client_type_e
 client.h, 494
 close
 TcpConnection, 271
 closeSocket
 SystemWrapper, 53
 CloseWindow
 IGraphicsLib, 125
 RayLib, 231
 cmd_func_t
 pending_cmd.h, 525
 command.h
 add_pending_cmd, 497
 broadcast, 498
 can_connect, 498
 can_start_incantation, 499
 check_player_starvation, 499
 check_rota_tiles, 499
 cleanup_client, 500
 cleanup_pending, 500
 cleanup_player_queue, 500
 command_bct, 500
 command_data_t, 497
 command_msz, 501
 command_mtc, 501
 command_pdi, 501
 command_pdr, 501
 command_pex, 502
 command_pkf, 502
 command_pgt, 502
 command_pie, 503
 command_pin, 503
 command_plv, 504
 command_ppo, 504
 command_sst, 504
 command_tna, 504
 connect_nbr, 505
 connect_nbr_srv, 505
 determine_type, 505
 eject, 506
 execute_com, 506
 find_client_by_id, 506
 finish_incantation, 506
 fork_c, 507
 forward, 507
 get_command_data, 508
 get_message, 508
 handle_player_death, 508
 inventory, 509
 left, 509
 look, 509
 MAX_LIFE_AFTER_FOOD, 497
 process_next_queued_command, 510
 right, 510
 send_bct_to_all_graphical_clients, 510
 send_enw_command, 510
 send_enw_command_start, 511
 send_mtc_to_all_graphical_clients, 511
 send_pnw_command_to_all, 511
 set_object, 512
 start_incantation, 512
 str_to_word_arr, 512
 take_object, 513
 tile_to_str, 513
 write_command_output, 513
 command_bct
 command.h, 500
 command_bct.c, 459
 command_bct.c
 calculate_size_bct_command, 459
 command_bct, 459
 command_mtc, 460
 get_buffer_bct_command, 460
 send_bct_command, 460
 send_bct_to_all_graphical_clients, 461
 send_mtc_to_all_graphical_clients, 461
 send_tile_content_to_one_client, 461
 command_data_s, 91
 accepted_types, 92
 commands, 92
 functions, 92
 times, 92
 command_data_t
 command.h, 497

command_ebo.c
 send_ebo_command, 462
command_edi.c
 send_edi_command, 463
command_enw.c
 get_buffer_for_enw, 464
 send_enw_command, 465
 send_enw_command_start, 465
 send_enw_command_to_client, 465
command_msz
 command.h, 501
 command_msz.c, 466
command_msz.c
 command_msz, 466
 send_msz_command, 466
command_mtc
 command.h, 501
 command_bct.c, 460
command_pbc
 command_pbc.c, 467
 graphical_commands.h, 516
command_pbc.c
 command_pbc, 467
command_pdi
 command.h, 501
 command_pdi.c, 468
command_pdi.c
 command_pdi, 468
command_pdr
 command.h, 501
 command_pdr.c, 469
command_pdr.c
 command_pdr, 469
command_pex
 command.h, 502
 command_pex.c, 470
command_pex.c
 command_pex, 470
command_pfk
 command.h, 502
 command_pfk.c, 471
command_pfk.c
 command_pfk, 471
command_pgt
 command.h, 502
 command_pgt.c, 472
command_pgt.c
 command_pgt, 472
command_pic
 command_pic.c, 474
 graphical_commands.h, 517
command_pic.c
 command_pic, 474
 get_pic_buffer, 474
command_pie
 command.h, 503
 command_pie.c, 475
command_pie.c
 command_pie, 475
 get_pie_buffer, 475
command_pin
 command.h, 503
 command_pin.c, 477
command_pin.c
 calculate_size_pin_command, 476
 command_pin, 477
 get_buffer_pin_command, 477
 send_pin_command, 477
 send_pin_to_all, 478
command_plv
 command.h, 504
 command_plv.c, 479
command_plv.c
 command_plv, 479
 send_plv_command, 479
 send_plv_to_all, 479
command_pnw.c
 calcul_size_pnw_command, 480
 send_all_player_info_to_one_client, 481
 send_pnw_command, 481
 send_pnw_command_to_all, 481
command_ppo
 command.h, 504
 command_ppo.c, 482
command_ppo.c
 command_ppo, 482
 find_client_by_id, 482
 get_ppo_buffer, 483
 send_ppo_command, 483
command_queue
 player_s, 198
command_seg
 command_seg.c, 484
 graphical_commands.h, 517
command_seg.c
 command_seg, 484
command_sgt
 command_sgt.c, 485
 graphical_commands.h, 518
command_sgt.c
 command_sgt, 485
command_smg.c
 send_smg_command, 486
command_sst
 command.h, 504
 command_sst.c, 487
command_sst.c
 command_sst, 487
 get_buffer_sst, 488
 get_time_from_buffer, 488
command_tna
 command.h, 504
 command_tna.c, 489
command_tna.c
 command_tna, 489
 send_one_tna_command, 489

send_tna_command, 489
commands
 command_data_s, 92
concatenate_message_parts
 broadcast.c, 418
connect
 NetworkManager, 184
 TcpConnection, 271
connect_nbr
 command.h, 505
 connect_nbr.c, 420
connect_nbr.c
 connect_nbr, 420
 connect_nbr_srv, 420
 count_team_eggs, 420
 count_team_players, 421
 format_response, 421
connect_nbr_srv
 command.h, 505
 connect_nbr.c, 420
Connection
 EggData, 102
connection.c
 add_fd, 564
 copy_names, 564
 create_server, 564
 init_new_client, 565
 init_server, 565
 init_server_resources, 566
 init_server_socket, 566
 remove_fd, 566
 remove_head_client, 566
 remove_other_client, 567
connection_err.c
 print_co, 568
 server_err, 568
connection_utils.c
 count_team, 569
 create_server_egg, 570
 init_server_eggs, 570
ConnectionCallback
 NetworkManager, 183
connectSocket
 SystemWrapper, 53
Constants.hpp
 DEFAULT_HEIGHT, 359
 DEFAULT_TITLE, 359
 DEFAULT_WIDTH, 360
 KEY_A, 358
 KEY_ALT_LEFT, 359
 KEY_B, 358
 KEY_BACKSPACE, 359
 KEY_C, 358
 KEY_CONTROL_LEFT, 359
 KEY_D, 358
 KEY_DELETE, 359
 KEY_DOWN, 358
 KEY_E, 358
 KEY_ENTER, 358
 KEY_ESCAPE, 358
 KEY_F, 358
 KEY_F1, 358
 KEY_F2, 358
 KEY_F3, 358
 KEY_G, 358
 KEY_H, 358
 KEY_I, 358
 KEY_INSERT, 359
 KEY_J, 358
 KEY_K, 358
 KEY_L, 358
 KEY_LEFT, 358
 KEY_M, 358
 KEY_N, 358
 KEY_NULL, 358
 KEY_O, 358
 KEY_P, 358
 KEY_Q, 358
 KEY_R, 358
 KEY_RIGHT, 358
 KEY_S, 358
 KEY_SHIFT_LEFT, 359
 KEY_SPACE, 358
 KEY_SUPER_LEFT, 359
 KEY_T, 358
 KEY_TAB, 359
 KEY_U, 358
 KEY_UP, 358
 KEY_V, 358
 KEY_W, 358
 KEY_X, 358
 KEY_Y, 358
 KEY_Z, 358
 KeyboardKey, 358
 NUM_RANDOM_MODELS, 360
convert_rotation_to_vector
 eject.c, 422
copy_names
 connection.c, 564
COUNT
 GraphicalContext.hpp, 362
 tile.h, 542
count
 circular_buffer_s, 87
count_team
 connection_utils.c, 569
 server.h, 534
count_team_eggs
 connect_nbr.c, 420
count_team_players
 connect_nbr.c, 421
count_valid_tokens
 array_function.c, 559
create_egg
 egg.c, 549
 egg.h, 515

create_map
 map_creation.c, 551
 server.h, 534
create_server
 connection.c, 564
 server.h, 535
create_server_egg
 connection_utils.c, 570
CreateCamera2D
 IGraphicsLib, 125
 RayLib, 232
CreateCamera3D
 IGraphicsLib, 126
 RayLib, 232
createDefaultConfig
 raylibcpp::Text3DWave, 284
createDefaultWaveConfig
 raylibcpp::Text3DConfig, 278
createDetailedTileStrategy
 Zappy::TileRenderStrategyFactory, 300
createFastWave
 raylibcpp::Text3DConfig, 278
createGraphicsLib
 RayLib.cpp, 343
createGuiLib
 RayGui.cpp, 317
createIntenseWave
 raylibcpp::Text3DConfig, 278
createModelTileStrategy
 Zappy::TileRenderStrategyFactory, 300
createShared
 Zappy::ModelManagerAdapter, 176
createSimpleTileStrategy
 Zappy::TileRenderStrategyFactory, 300
createSlowWave
 raylibcpp::Text3DConfig, 278
createSocket
 SystemWrapper, 53
createSubtleWave
 raylibcpp::Text3DConfig, 278
createWaveConfig
 raylibcpp::Text3DConfig, 279
CRITICAL
 logger.LogLevel, 155
critical
 logger.Logger, 153
cross
 raylibcpp::Vector3Wrap, 309
current_behaviour
 agent.Agent, 72
current_resources
 server_s, 257
current_tick
 server_s, 257
data
 SystemWrapper::SafeBuffer, 250
Death
 EggData, 102

DEBUG
 logger.LogLevel, 156
debug
 logger.Logger, 153
decisionManager, 43
 agent.Agent, 72
decisionManager.DecisionManager, 92
 __init__, 92
 agent, 93
 behaviors, 93
 take_action, 93
decrypt_message
 encryption, 43
DEFAULT_HEIGHT
 Constants.hpp, 359
DEFAULT_TITLE
 Constants.hpp, 359
DEFAULT_WIDTH
 Constants.hpp, 360
dequeue
 MessageQueue, 167
dequeueWait
 MessageQueue, 168
DERAUMERE
 GraphicalContext.hpp, 362
 tile.h, 542
DERAUMERE_D
 tile.h, 540
DetailedTileRenderStrategy
 Zappy::DetailedTileRenderStrategy, 94
determine_type
 command.h, 505
 take_object.c, 452
direction_push
 eject.c, 422
disconnect
 NetworkManager, 185
disp_args
 main.c, 545
display_help
 help.c, 490
 server.h, 535
displayHelp
 main.cpp, 363
 ParsingCLI, 195
distance
 CameraController, 83
distribute_one_resource
 resources_map.c, 552
distribute_resources
 resources_map.c, 553
 tile.h, 542
DLLoader
 DLLoader< T >, 96
DLLoader< T >, 95
 ~DLLoader, 96
 DLLoader, 96
 getInstance, 96

getSymbol, 97
DOWN
player.h, 528
Draw
RayGUICPP::Button, 80
RayGUICPP::CheckBox, 87
RayGUICPP::ColorPicker, 90
RayGUICPP::ComboBox, 91
RayGUICPP::DropdownBox, 98
RayGUICPP::GroupBox, 119
RayGUICPP::Label, 149
RayGUICPP::ListView, 152
RayGUICPP::MessageBox, 166
RayGUICPP::Panel, 188
RayGUICPP::ProgressBar, 209
RayGUICPP::ScrollPane, 255
RayGUICPP::Slider, 264
RayGUICPP::Spinner, 269
RayGUICPP::StatusBar, 270
RayGUICPP::TabBar, 270
RayGUICPP::TextBox, 285
RayGUICPP::TextInputBox, 285
RayGUICPP::Toggle, 302
RayGUICPP::ToggleGroup, 303
RayGUICPP::ValueBox, 307
RayGUICPP::WindowBox, 313
draw
raylibcpp::ModelWrap, 180
raylibcpp::Text3DCodepoint, 276
raylibcpp::Texture, 286
DrawButton
IGuiLib, 133
RayGui, 220
DrawCheckBox
IGuiLib, 134
RayGui, 220
DrawCircle
IGraphicsLib, 126
RayLib, 232
drawCircle
raylibcpp::Shape, 260
DrawColorPicker
IGuiLib, 134
RayGui, 221
DrawComboBox
IGuiLib, 134
RayGui, 221
DrawCube
IGraphicsLib, 126
RayLib, 232
drawCube
raylibcpp::Shape, 261
DrawCylinder
IGraphicsLib, 126
RayLib, 233
drawCylinder
raylibcpp::Shape, 261
DrawDropdownBox
IGuiLib, 134
RayGui, 220
DrawGrid
IGraphicsLib, 126
RayLib, 233
drawGrid
raylibcpp::Shape, 261
DrawGroupBox
IGuiLib, 134
RayGui, 222
DrawLabel
IGuiLib, 135
RayGui, 222
DrawLine3D
IGraphicsLib, 126
RayLib, 233
drawLine3D
raylibcpp::Shape, 261
DrawListView
IGuiLib, 135
RayGui, 222
DrawMessageBox
IGuiLib, 135
RayGui, 222
drawModel
ModelManager, 171
DrawModel3D
IGraphicsLib, 127
RayLib, 233
DrawModelEx
IGraphicsLib, 127
RayLib, 234
drawModelEx
ModelManager, 171
DrawPanel
IGuiLib, 135
RayGui, 223
DrawPlane
IGraphicsLib, 127
RayLib, 234
drawPlane
raylibcpp::Shape, 261
DrawProgressBar
IGuiLib, 135
RayGui, 223
DrawRectangle
IGraphicsLib, 127
RayLib, 234
drawRectangle
raylibcpp::Shape, 262
DrawScrollPane
IGuiLib, 136
RayGui, 223
DrawSlider
IGuiLib, 136
RayGui, 223
DrawSphere
IGraphicsLib, 127

RayLib, 234
drawSphere
 raylibcpp::Shape, 262
DrawSpinner
 IGuiLib, 136
 RayGui, 224
DrawStatusBar
 IGuiLib, 136
 RayGui, 224
DrawTabBar
 IGuiLib, 137
 RayGui, 224
DrawText
 IGraphicsLib, 127
 RayLib, 235
drawText
 raylibcpp::Font, 106
DrawText3D
 IGraphicsLib, 128
 RayLib, 235
 raylibcpp::Text3D, 275
DrawTextBox
 IGuiLib, 137
 RayGui, 225
DrawTextCodepoint3D
 raylibcpp::Text3D, 275
DrawTextInputBox
 IGuiLib, 137
 RayGui, 225
drawTexture
 TextureManager, 290
DrawTexture2D
 IGraphicsLib, 128
 RayLib, 235
DrawTextWave3D
 IGraphicsLib, 128
 RayLib, 235
 raylibcpp::Text3D, 275
DrawToggle
 IGuiLib, 137
 RayGui, 225
DrawToggleGroup
 IGuiLib, 137
 RayGui, 225
DrawValueBox
 IGuiLib, 138
 RayGui, 226
DrawWindowBox
 IGuiLib, 138
 RayGui, 226
Drop
 EggData, 102
EBO_HEADER
 HeaderMessage.hpp, 371
EDI_HEADER
 HeaderMessage.hpp, 371
Egg
 IMessageData.hpp, 378
egg.c
 add_egg, 548
 add_egg_to_tile, 549
 create_egg, 549
 remove_egg, 549
 remove_egg_by_id, 550
 remove_egg_from_tile, 550
 remove_id_from_array, 550
 remove_node, 550
egg.h
 add_egg, 514
 create_egg, 515
 egg_t, 514
 remove_egg, 515
 server_t, 514
 tile_t, 514
 egg_capacity
 tile_s, 293
 egg_count
 tile_s, 293
 egg_id
 egg_s, 100
 egg_ids
 tile_s, 293
 egg_s, 99
 egg_id, 100
 next, 100
 pos_x, 100
 pos_y, 100
 team_name, 100
 egg_t
 egg.h, 514
EggAction
 EggData, 102
EggData, 101
 Connection, 102
 Death, 102
 Drop, 102
 EggAction, 102
 EggData, 102
 getAction, 102
 getEggId, 102
 getPlayerId, 103
 getType, 103
 getX, 103
 getY, 103
 Laying, 102
 setEggId, 103
 setPlayerId, 103
 setX, 103
 setY, 103
eggIds
 TileData, 299
eggs
 server_s, 257
eject
 command.h, 506
 eject.c, 423

eject.c
 convert_rotation_to_vector, 422
 direction_push, 422
 eject, 423
 get_string_to_send, 423
 push_client, 423
 push_eggs, 424
 push_single_client, 424
 wrap_position, 424
 encrypt_message
 encryption, 44
 encryption, 43
 decrypt_message, 43
 encrypt_message, 44
 secret_key, 44
 end
 circular_buffer_s, 88
 EndCamera3D
 IGraphicsLib, 128
 RayLib, 236
 EndDrawing
 IGraphicsLib, 128
 RayLib, 236
 endDrawing
 raylibcpp::Window, 312
 EndGame
 IMessageData.hpp, 378
 EndGameData, 104
 EndGameData, 105
 getTeamName, 105
 getType, 105
 setTeamName, 105
 endMode
 raylibcpp::Camera3DWrap, 82
 enqueue
 MessageQueue, 168
 ENW_HEADER
 HeaderMessage.hpp, 371
 ERROR
 logger.LogLevel, 156
 error
 logger.Logger, 153
 execute
 behaviors.Behavior, 76
 behaviors.DysonBehavior, 99
 behaviors.GetFoodAndMineralsBehavior, 110
 behaviors.GetFoodBehavior, 111
 behaviors.GetMineralsBehavior, 113
 behaviors.UpgradeBehavior, 306
 execute_com
 command.h, 506
 parse_command.c, 439
 execute_graphical_command
 parse_command.c, 439
 execute_if_free
 parse_command.c, 439
 execute_pending_cmd
 pending_cmd_utils.c, 445
 pending_cmd_utils.h, 526
 executeWithTimeout
 SystemWrapper, 53
 extract_command
 circular_buffer.c, 415
 circular_buffer.h, 492
 extractCommandParameter
 ProtocolParser, 211
 extractMessageParameters
 ProtocolParser, 211
 fds
 poll_manager_s, 208
 file
 logger.Logger, 154
 file_handle
 logger.Logger, 154
 fill_poll_array
 server_run.c, 576
 fill_result_array
 array_function.c, 559
 find_and_execute
 parse_command.c, 440
 find_client_by_id
 command.h, 506
 command_ppo.c, 482
 find_client_by_socket
 server_run.c, 576
 find_command_end
 circular_buffer.c, 415
 circular_buffer.h, 492
 find_egg_for_team
 player_init.c, 586
 finish_incantation
 command.h, 506
 finish_incantation.c, 426
 server.h, 535
 finish_incantation.c
 check_win, 425
 finish_incantation, 426
 handle_incantation_failure, 426
 handle_incantation_success, 426
 handle_win, 427
 remove_resources, 427
 Font
 raylibcpp::Font, 105
 FOOD
 GraphicalContext.hpp, 362
 tile.h, 542
 FOOD_D
 tile.h, 540
 fork.c
 fork_c, 428
 get_next_egg_id, 428
 fork_c
 command.h, 507
 fork.c, 428
 format_look
 look.c, 433

format_response
 connect_nbr.c, 421

format_str_resources
 tile_to_str.c, 454

format_tile
 look.c, 434

forward
 command.h, 507
 forward.c, 430

forward.c
 change_map_pos, 429
 forward, 430
 wrap_position, 430

fps
 UIRenderer, 304

free.c
 free_all, 456
 free_arr, 456
 free_client, 457
 free_eggs, 457
 free_map, 457
 free_parsed_info, 458
 free_poll_mana, 458
 free_tiles, 458

free_all
 free.c, 456
 server.h, 536

free_arr
 free.c, 456
 parsing.h, 523

free_client
 free.c, 457

free_cpy_arr
 array_function.c, 560

free_eggs
 free.c, 457

free_inventory
 player.h, 529
 player_inventory.c, 588

free_map
 free.c, 457

free_names
 parsing_err.c, 583

free_node
 client_node.c, 562
 server.h, 536

free_parsed_info
 free.c, 458

free_poll_mana
 free.c, 458

free_tiles
 free.c, 458

frequency
 parsing_info_s, 192

FromRaylib
 TypeAdapter, 56, 57

func
 pending_cmd_s, 197

functions
 command_data_s, 92

g
 ZappyTypes::Color, 90

GameLoop, 106
 ~GameLoop, 107
 GameLoop, 107
 init, 108
 onMapSizeChanged, 108
 onTileChanged, 108
 renderCube, 108
 run, 108
 setServerInfo, 109

get
 NetworkLogger, 182
 raylibcpp::Camera2DWrap, 80
 raylibcpp::Camera3DWrap, 82
 raylibcpp::Font, 106
 raylibcpp::ModelWrap, 180
 raylibcpp::MusicWrap, 181
 raylibcpp::SoundWrap, 268
 raylibcpp::Texture, 286
 raylibcpp::Texture3DWrap, 288
 raylibcpp::Vector3Wrap, 310

get_base_direction
 broadcast.c, 418

get_best_available_resource
 zappy, 58

get_buffer_bct_command
 command_bct.c, 460

get_buffer_for_enw
 command_enw.c, 464

get_buffer_pin_command
 command_pin.c, 477

get_buffer_sst
 command_sst.c, 488

get_client_by_id_player
 start_incantation.c, 449

get_closest_of_item
 zappy, 58

get_command_data
 command.h, 508
 parse_command.c, 440

get_inventory_content
 player.h, 529
 player_inventory_display.c, 590

get_item_relative_pos
 zappy, 59

get_key
 input, 44

get_message
 agent.Agent, 70
 command.h, 508
 parse_command.c, 440
 socketManager.SocketManager, 265

get_next_egg_id
 fork.c, 428

get_one_item_content

player_inventory_display.c, 590
get_pic_buffer
 command_pic.c, 474
get_pie_buffer
 command_pie.c, 475
get_ppo_buffer
 command_ppo.c, 483
get_resource_name
 player.h, 529
 player_inventory_utils.c, 591
get_running_flag
 main.c, 546
get_string_to_send
 eject.c, 423
get_time_from_buffer
 command_sst.c, 488
get_total_size
 look.c, 434
get_total_size_tile
 look.c, 434
getAction
 EggData, 102
getAddr
 SystemWrapper::SafeSockAddr, 254
getBoundingBox
 ModelManager, 171
getCommandName
 ProtocolParser, 211
getData
 Message, 165
getDeraumere
 PlayerInventoryData, 205
 TileContentData, 296
getDominantResourceType
 GraphicalContext, 116
getEggId
 EggData, 102
getErrorString
 SystemWrapper, 54
getEvents
 SystemWrapper::SafePollFd, 252
getFd
 SystemWrapper::SafePollFd, 252
getFlagValue
 paringArgsClass.parseArgs, 189
getFood
 PlayerInventoryData, 205
 TileContentData, 296
getFormattedMessage
 AException, 68
 IException, 122
getGlyphIndex
 raylibcpp::Text3DHelper, 50
getGraphicsLib
 LibraryManager, 149
getGraphicsLibPtr
 LibraryManager, 150
getGuiLib
 LibraryManager, 150
getGuiLibPtr
 LibraryManager, 150
getHeader
 Message, 165
getHeight
 MapSizeData, 162
getId
 PlayerInfoData, 201
 PlayerInventoryData, 205
getInstance
 DLLoader< T >, 96
 LibraryManager, 150
 ModelManager, 172
 TextureManager, 290
getLastError
 LibraryManager, 151
getLevel
 IncantationData, 142
 PlayerInfoData, 201
getLinemate
 PlayerInventoryData, 205
 TileContentData, 296
getMachine
 paringArgsClass.parseArgs, 189
 ParsingCLI, 195
getManager
 Zappy::ModelManagerAdapter, 176
getMapHeight
 GraphicalContext, 117
 Renderer, 244
getMapWidth
 GraphicalContext, 117
 Renderer, 244
getMendiane
 PlayerInventoryData, 205
 TileContentData, 296
getMessage
 AException, 69
 BroadcastData, 77
 IException, 122
 Message, 165
 ServerMessageData, 260
getModelCount
 ModelManager, 172
getModelId
 ModelManager, 172
GetMousePosition
 IGraphicsLib, 128
 RayLib, 236
getMousePosition
 raylibcpp::Input, 146
GetMouseWheelMove
 IGraphicsLib, 129
 RayLib, 236
getMouseWheelMove
 raylibcpp::Input, 146
GetMouseX

IGraphicsLib, 129
RayLib, 237
getMouseX
 raylibcpp::Input, 146
GetMouseY
 IGraphicsLib, 129
 RayLib, 237
getMouseY
 raylibcpp::Input, 146
getName
 paringArgsClass.parseArgs, 189
getNames
 TeamNameData, 274
getNextId
 raylibcpp::Texture, 287
getOrientation
 PlayerInfoData, 201
getPhiras
 PlayerInventoryData, 205
 TileContentData, 296
getPlayerId
 BroadcastData, 77
 EggData, 103
 ResourceData, 249
getPlayerIds
 IncantationData, 142
getPlayerInfo
 GraphicalContext, 117
getPollFd
 SystemWrapper::SafePollFd, 252
getPollPtr
 SystemWrapper::SafePollFd, 252, 253
getPort
 paringArgsClass.parseArgs, 190
 ParsingCLI, 195
getResourceQuantity
 GraphicalContext, 117
getResourceTextureId
 Renderer, 244
getResourceType
 ResourceData, 249
getRevents
 SystemWrapper::SafePollFd, 253
getSibur
 PlayerInventoryData, 205
 TileContentData, 296
getSize
 SystemWrapper::SafeSockAddr, 254
getSockAddr
 SystemWrapper::SafeSockAddr, 255
getSocketOption
 SystemWrapper, 54
getStructuredData
 Message, 165
GetStyle
 IGuiLib, 138
 RayGui, 226
 RayGUICPP::RayGui, 228
getSymbol
 DLLoader< T >, 97
getTeamName
 EndGameData, 105
 PlayerInfoData, 202
getTextureCount
 TextureManager, 290
getTextureId
 TextureManager, 290
getThystame
 PlayerInventoryData, 206
 TileContentData, 296
getTileData
 GraphicalContext, 118
getTimeUnit
 TimeUnitData, 301
getType
 AException, 69
 BroadcastData, 77
 EggData, 103
 EndGameData, 105
 IException, 122
 IMessageData, 141
 IncantationData, 142
 IncantationEndData, 145
 MapSizeData, 162
 PlayerInfoData, 202
 PlayerInventoryData, 206
 ResourceData, 249
 ServerMessageData, 260
 TeamNameData, 274
 TileContentData, 296
 TimeUnitData, 302
getWidth
 MapSizeData, 162
getX
 EggData, 103
 IncantationData, 143
 IncantationEndData, 145
 PlayerInfoData, 202
 PlayerInventoryData, 206
 TileContentData, 297
getY
 EggData, 103
 IncantationData, 143
 IncantationEndData, 145
 PlayerInfoData, 202
 PlayerInventoryData, 206
 TileContentData, 297
go_get_item
 zappy, 59
go_take_item
 actions, 41
go_to_pos_with_distance
 actions, 41
got_to_dir
 actions, 42
GRAPHICAL

client.h, 494
graphical_client.c
 add_graphic_client, 571
 add_to_graphical_list, 571
 remove_graphic_client, 572
 remove_graphical_node, 572
 send_map_info_to_one_client, 572
 send_message_to_all_graphic, 572
graphical_client_s, 114
 client, 114
 next, 114
graphical_client_t
 client.h, 494
graphical_clients
 server_s, 257
graphical_commands.h
 command_pbc, 516
 command_pic, 517
 command_seg, 517
 command_sgt, 518
 send_all_player_info_to_one_client, 518
 send_ebo_command, 518
 send_edi_command, 519
 send_msz_command, 519
 send_pin_command, 519
 send_pin_to_all, 519
 send_plv_command, 520
 send_plv_to_all, 520
 send_pnw_command, 520
 send_ppo_command, 521
 send_smg_command, 521
 send_team_names_to_one_client, 521
 send_tile_content_to_one_client, 522
 send_tna_command, 522
GraphicalContext, 114
 ~GraphicalContext, 115
 addObserver, 116
 getDominantResourceType, 116
 getMapHeight, 117
 getMapWidth, 117
 getPlayerInfo, 117
 getResourceQuantity, 117
 getTileData, 118
 GraphicalContext, 115
 isPlayerOnTile, 118
 removeObserver, 118
 updateContext, 119
 UpdateFunction, 115
GraphicalContext.hpp
 COUNT, 362
 DERAUMERE, 362
 FOOD, 362
 LINEMATE, 362
 MENDIANE, 362
 PHIRAS, 362
 ResourceType, 362
 SIBUR, 362
 THYSTAME, 362
 handle_game_tick
 server_run.c, 577
 handle_incantation_failure
 finish_incantation.c, 426
 handle_incantation_success
 finish_incantation.c, 426
 handle_player_death
 command.h, 508
 player_death.c, 584
 handle_win
 finish_incantation.c, 427
handleKeyboardInput
 CameraController, 83
handleMouseInput
 CameraController, 84
handleProtocol
 ProtocolParser, 211
has_enough_resources
 start_incantation.c, 450
has_messages
 agent.Agent, 71
 socketManager.SocketManager, 265
hasData
 TcpConnection, 272
hasFlagAndValue
 paringArgsClass.parseArgs, 190
hasModel
 ModelManager, 173
hasModelLibSet
 ModelManager, 173
hasStructuredData
 Message, 165
hasTexture
 TextureManager, 291
hasTextureLibSet
 TextureManager, 291
HeaderMessage.hpp
 BCT_HEADER, 371
 EBO_HEADER, 371
 EDI_HEADER, 371
 ENW_HEADER, 371
 MCT_HEADER, 371
 MSZ_HEADER, 371
 PBC_HEADER, 371
 PDI_HEADER, 372
 PDR_HEADER, 372
 PEX_HEADER, 372
 PFK_HEADER, 372
 PGT_HEADER, 372
 PIC_HEADER, 372
 PIE_HEADER, 372
 PIN_HEADER, 372
 PLV_HEADER, 372
 PNW_HEADER, 372
 PPO_HEADER, 373
 SBP_HEADER, 373
 SEG_HEADER, 373
 SGT_HEADER, 373

SMG_HEADER, 373
SST_HEADER, 373
SUC_HEADER, 373
TNA_HEADER, 373
height
 parsing_info_s, 192
 raylibcpp::GlyphDimensions, 113
 resource_dist_s, 247
 ZappyTypes::Rectangle, 242
help.c
 display_help, 490
hexToColor
 raylibcpp::Utils, 307
how_many_in_inventory
 player.h, 529
 player_inventory_utils.c, 592
how_many_players_needed
 start_incantation.c, 450
how_much_of_item_here
 zappy, 59

IconText
 IGuiLib, 138
 RayGui, 226
 RayGUICPP::Icon, 120
id
 agent.Agent, 73
IException, 120
 ~IException, 121
 getFormattedMessage, 122
 getMessage, 122
 getType, 122
 what, 122
IGraphicalContextObserver, 122
 ~IGraphicalContextObserver, 123
 onMapSizeChanged, 123
 onTileChanged, 123
IGraphicsLib, 123
 ~IGraphicsLib, 125
 BeginCamera3D, 125
 BeginDrawing, 125
 BindTexture3D, 125
 ClearBackground, 125
 CloseWindow, 125
 CreateCamera2D, 125
 CreateCamera3D, 126
 DrawCircle, 126
 DrawCube, 126
 DrawCylinder, 126
 DrawGrid, 126
 DrawLine3D, 126
 DrawModel3D, 127
 DrawModelEx, 127
 DrawPlane, 127
 DrawRectangle, 127
 DrawSphere, 127
 DrawText, 127
 DrawText3D, 128
 DrawTexture2D, 128
DrawTextWave3D, 128
EndCamera3D, 128
EndDrawing, 128
GetMousePosition, 128
GetMouseWheelMove, 129
GetMouseX, 129
GetMouseY, 129
InitWindow, 129
IsKeyDown, 129
IsKeyPressed, 129
IsMouseButtonDown, 129
IsMouseButtonPressed, 129
IsMouseButtonReleased, 129
IsTextureReady, 130
LoadFont, 130
LoadModel3D, 130
LoadModelWithTexture, 130
LoadTexture2D, 130
LoadTexture3D, 130
MeasureText3D, 130
PlayMusic, 130
PlaySound, 131
setFps, 131
SetMusicVolume, 131
SetSoundVolume, 131
StopMusic, 131
StopSound, 131
UnloadFont, 131
UnloadModel3D, 131
UnloadTexture2D, 131
UnloadTexture3D, 132
UpdateMusic, 132
WindowShouldClose, 132
IGuiLib, 132
 ~IGuiLib, 133
 ButtonPressed, 133
 DrawButton, 133
 DrawCheckBox, 134
 DrawColorPicker, 134
 DrawComboBox, 134
 DrawDropdownBox, 134
 DrawGroupBox, 134
 DrawLabel, 135
 DrawListView, 135
 DrawMessageBox, 135
 DrawPanel, 135
 DrawProgressBar, 135
 DrawScrollPane, 136
 DrawSlider, 136
 DrawSpinner, 136
 DrawStatusBar, 136
 DrawTabBar, 137
 DrawTextBox, 137
 DrawTextInputBox, 137
 DrawToggle, 137
 DrawToggleGroup, 137
 DrawValueBox, 138
 DrawWindowBox, 138

GetStyle, 138
IconText, 138
LoadStyle, 138
SetStyle, 138
IMessageData, 139
~IMessageData, 141
getType, 141
IMessageData.hpp
Broadcast, 378
Egg, 378
EndGame, 378
Incantation, 378
IncantationEnd, 378
MapSize, 378
MessageType, 378
PlayerInfo, 378
PlayerInventory, 378
Resource, 378
ServerMessage, 378
TeamName, 378
TileContent, 378
TimeUnit, 378
Incantation
IMessageData.hpp, 378
incantation_leader_id
player_s, 198
IncantationData, 141
addPlayerId, 142
getLevel, 142
getPlayerIds, 142
getType, 142
getX, 143
getY, 143
IncantationData, 142
setLevel, 143
setPlayerIds, 143
setX, 143
setY, 143
IncantationEnd
IMessageData.hpp, 378
IncantationEndData, 143
getType, 145
getX, 145
getY, 145
IncantationEndData, 144
isSuccess, 145
setSuccess, 145
setX, 145
setY, 145
inetNtop
SystemWrapper, 54
inetPton
SystemWrapper, 54
INFO
logger.LogLevel, 156
info
logger.Logger, 154
init
CameraController, 85
GameLoop, 108
Renderer, 244
init_circular_buffer
circular_buffer.c, 415
circular_buffer.h, 492
init_inventory
player.h, 530
player_inventory.c, 588
init_new_client
connection.c, 565
init_new_player_pos
player_init.c, 586
server.h, 537
init_pending
player_init.c, 586
init_player
player.h, 530
player_init.c, 586
init_player_team
player_init.c, 587
init_resource_dist
resources_map.c, 553
init_result_array
array_function.c, 560
init_server
connection.c, 565
init_server_eggs
connection_utils.c, 570
server.h, 537
init_server_resources
connection.c, 566
init_server_socket
connection.c, 566
initialize
Zappy::MapRenderer, 157
InitWindow
IGraphicsLib, 129
RayLib, 237
input, 44
get_key, 44
inventory
command.h, 509
inventory.c, 431
player_s, 199
inventory.c
inventory, 431
inventory_has_item
player.h, 530
player_inventory_utils.c, 592
inventory_size
player_s, 199
ip
agent.Agent, 73
main, 46
is_dup_flag
parsing.c, 580
is_fully_connected

client_s, 89
is_in_incantation
 player_s, 199
is_valid_team_name
 parse_command_utils_bis.c, 444
isConnected
 NetworkManager, 185
 TcpConnection, 272
isDragging
 CameraController, 85
isEmpty
 MessageQueue, 168
isEnabled
 NetworkLogger, 182
isIncantating
 TileData, 299
IsKeyDown
 IGraphicsLib, 129
 RayLib, 237
isKeyDown
 raylibcpp::Input, 146
IsKeyPressed
 IGraphicsLib, 129
 RayLib, 237
isKeyPressed
 raylibcpp::Input, 147
IsMouseButtonDown
 IGraphicsLib, 129
 RayLib, 237
isMouseButtonDown
 raylibcpp::Input, 147
IsMouseButtonPressed
 IGraphicsLib, 129
 RayLib, 238
isMouseButtonPressed
 raylibcpp::Input, 147
IsMouseButtonReleased
 IGraphicsLib, 129
 RayLib, 238
isMouseButtonReleased
 raylibcpp::Input, 147
isNewline
 raylibcpp::Text3DHelper, 50
isPlayerOnTile
 GraphicalContext, 118
isReady
 raylibcpp::Texture, 287
isRenderableCharacter
 raylibcpp::Text3DHelper, 50
isRunning
 NetworkThread, 187
isShowLetterBoundary
 raylibcpp::Text3DConfig, 279
isShowTextBoundary
 raylibcpp::Text3DConfig, 279
 isSuccess
 IncantationEndData, 145
IsTextureReady

IGraphicsLib, 130
RayLib, 238
isValidConfig
 raylibcpp::Text3DWave, 284
isValidFont
 raylibcpp::Text3DHelper, 50
isValidRenderParams
 raylibcpp::Text3DHelper, 50
isValidWaveConfig
 raylibcpp::Text3DConfig, 279
isWaveMarker
 raylibcpp::Text3DParser, 281
isWhitespace
 raylibcpp::Text3DHelper, 51

KEY_A
 Constants.hpp, 358
KEY_ALT_LEFT
 Constants.hpp, 359
KEY_B
 Constants.hpp, 358
KEY_BACKSPACE
 Constants.hpp, 359
KEY_C
 Constants.hpp, 358
KEY_CONTROL_LEFT
 Constants.hpp, 359
KEY_D
 Constants.hpp, 358
KEY_DELETE
 Constants.hpp, 359
KEY_DOWN
 Constants.hpp, 358
KEY_E
 Constants.hpp, 358
KEY_ENTER
 Constants.hpp, 358
KEY_ESCAPE
 Constants.hpp, 358
KEY_F
 Constants.hpp, 358
KEY_F1
 Constants.hpp, 358
KEY_F2
 Constants.hpp, 358
KEY_F3
 Constants.hpp, 358
KEY_G
 Constants.hpp, 358
KEY_H
 Constants.hpp, 358
KEY_I
 Constants.hpp, 358
KEY_INSERT
 Constants.hpp, 359
KEY_J
 Constants.hpp, 358
KEY_K
 Constants.hpp, 358

KEY_L
 Constants.hpp, 358

KEY_LEFT
 Constants.hpp, 358

KEY_M
 Constants.hpp, 358

KEY_N
 Constants.hpp, 358

KEY_NULL
 Constants.hpp, 358

KEY_O
 Constants.hpp, 358

KEY_P
 Constants.hpp, 358

KEY_Q
 Constants.hpp, 358

KEY_R
 Constants.hpp, 358

KEY_RIGHT
 Constants.hpp, 358

KEY_S
 Constants.hpp, 358

KEY_SHIFT_LEFT
 Constants.hpp, 359

KEY_SPACE
 Constants.hpp, 358

KEY_SUPER_LEFT
 Constants.hpp, 359

KEY_T
 Constants.hpp, 358

KEY_TAB
 Constants.hpp, 359

KEY_U
 Constants.hpp, 358

KEY_UP
 Constants.hpp, 358

KEY_V
 Constants.hpp, 358

KEY_W
 Constants.hpp, 358

KEY_X
 Constants.hpp, 358

KEY_Y
 Constants.hpp, 358

KEY_Z
 Constants.hpp, 358

KeyboardKey
 Constants.hpp, 358

ZappyTypes, 61

Laying
 EggData, 102

LEFT
 player.h, 528

left
 command.h, 509

left.c, 432

left.c
 change_rot, 432

left, 432

LETTER_BOUNDARY_COLOR
 Text3DCore.hpp, 331

LETTER_BOUNDARY_SIZE
 Text3DCore.hpp, 331

level
 agent.Agent, 73

player_s, 199

LibraryManager, 149

getGraphicsLib, 149

getGraphicsLibPtr, 150

getGuiLib, 150

getGuiLibPtr, 150

getInstance, 150

getLastErrorHandler, 151

loadGraphicsLib, 151

loadGuiLib, 151

life
 player_s, 199

LINEMATE
 GraphicalContext.hpp, 362

tile.h, 542

LINEMATE_D
 tile.h, 541

LoadFont
 IGraphicsLib, 130

RayLib, 238

loadGraphicsLib
 LibraryManager, 151

loadGuiLib
 LibraryManager, 151

loadModel
 ModelManager, 173

LoadModel3D
 IGraphicsLib, 130

RayLib, 238

LoadModelWithTexture
 IGraphicsLib, 130

RayLib, 239

loadModelWithTextures
 ModelManager, 174

loadResourceTexture
 Renderer, 244

LoadStyle
 IGuiLib, 138

RayGui, 227

RayGUICPP::RayGui, 228

loadTexture
 TextureManager, 291

LoadTexture2D
 IGraphicsLib, 130

RayLib, 239

LoadTexture3D
 IGraphicsLib, 130

RayLib, 239

lock
 socketManager.SocketManager, 266

log

logger.Logger, 154
NetworkLogger, 182
logger, 45
 agent.Agent, 73
logger.Logger, 153
 __init__, 153
 critical, 153
 debug, 153
 error, 153
 file, 154
 file_handle, 154
 info, 154
 log, 154
 message_prefix, 154
 warning, 154
logger.LogLevel, 155
 CRITICAL, 155
 DEBUG, 156
 ERROR, 156
 INFO, 156
 WARNING, 156
look
 command.h, 509
 look.c, 435
look.c
 format_look, 433
 format_tile, 434
 get_total_size, 434
 get_total_size_tile, 434
 look, 435
 look_tiles, 435
look_tiles
 look.c, 435
look_utils.c
 check_down, 436
 check_left, 436
 check_right, 437
 check_rota_tiles, 437
 check_up, 437
machine
 paringArgsClass.parseArgs, 191
main, 45
 args, 45
 child_pids, 46
 ip, 46
 main.c, 546
 main.cpp, 363
 num_agents, 46
 pid, 46
 port, 46
 run_agent, 45
 signal_handler, 45
 team, 46
main.c
 check_mini_args, 545
 disp_args, 545
 get_running_flag, 546
 main, 546
server_loop, 546
set_rdm_seed, 547
setup_signal_handler, 547
should_continue_running, 547
sigint_handler, 547
sigpipe_handler, 548
main.cpp
 displayHelp, 363
 main, 363
makeSharedArray
 SystemWrapper, 54
makeUniqueArray
 SystemWrapper, 54
manage_broadcast
 broadcastManager.BroadcastManager, 78
map
 resource_dist_s, 247
 server_s, 257
map_creation.c
 create_map, 551
map_size_x
 agent.Agent, 73
map_size_y
 agent.Agent, 73
MapRenderer
 Zappy::MapRenderer, 157
MapRendererObserver
 Zappy::MapRendererObserver, 160
MapSize
 IMessageData.hpp, 378
MapSizeData, 161
 getHeight, 162
 getType, 162
 getWidth, 162
 MapSizeData, 162
 setHeight, 163
 setWidth, 163
MAX_FONT_SIZE
 raylibcpp::Text3DHelper, 51
MAX_LIFE_AFTER_FOOD
 command.h, 497
MAX_SPACING
 raylibcpp::Text3DHelper, 51
MCT_HEADER
 HeaderMessage.hpp, 371
measureText
 raylibcpp::Text3DMeasurement, 280
MeasureText3D
 IGraphicsLib, 130
 RayLib, 239
MeasureTextWave3D
 raylibcpp::Text3D, 276
MENDIANE
 GraphicalContext.hpp, 362
 tile.h, 542
MENDIANE_D
 tile.h, 541
Message, 163

~Message, 164
 addIntParam, 164
 addStringParam, 164
 getData, 165
 getHeader, 165
 getMessage, 165
 getStructuredData, 165
 hasStructuredData, 165
 Message, 164
 setData, 165
 setHeader, 166
 setMessage, 166
message
 paringArgsClass.ParsingArgsError, 194
message_prefix
 logger.Logger, 154
message_queue
 socketManager.SocketManager, 267
MessageCallback
 NetworkManager, 183
messageComplete
 ProtocolParser, 211
MessageQueue, 167
 ~MessageQueue, 167
 clear, 167
 dequeue, 167
 dequeueWait, 168
 enqueue, 168
 isEmpty, 168
 MessageQueue, 167
 size, 168
MessageType
 IMessageData.hpp, 378
MIN_FONT_SIZE
 raylibcpp::Text3DHelper, 51
MIN_SPACING
 raylibcpp::Text3DHelper, 51
Model3D, 169
 boundingBoxMax, 169
 boundingBoxMin, 169
 modelId, 169
 scale, 169
 textureIds, 170
modelId
 Model3D, 169
ModelManager, 170
 drawModel, 171
 drawModelEx, 171
 getBoundingBox, 171
 getInstance, 172
 getModelCount, 172
 getModelId, 172
 hasModel, 173
 hasModelLibSet, 173
 loadModel, 173
 loadModelWithTextures, 174
 setGraphicsLib, 175
 unloadAllModels, 175
 unloadModel, 175
ModelManagerAdapter
 Zappy::ModelManagerAdapter, 176
ModelTileRenderStrategy
 Zappy::ModelTileRenderStrategy, 178
ModelWrap
 raylibcpp::ModelWrap, 179
MSZ_HEADER
 HeaderMessage.hpp, 371
MusicWrap
 raylibcpp::MusicWrap, 180
name
 paringArgsClass.parseArgs, 191
 player_inventory_s, 197
names
 parsing_info_s, 192
nb_valid_players_on_tile
 start_incantation.c, 450
needs_rebuild
 poll_manager_s, 208
NetworkLogger, 182
 get, 182
 isEnabled, 182
 log, 182
 setEnabled, 183
NetworkManager, 183
 ~NetworkManager, 184
 connect, 184
 ConnectionCallback, 183
 disconnect, 185
 isConnected, 185
 MessageCallback, 183
 NetworkManager, 184
 networkThreadLoop, 185
 processIncomingMessages, 185
 sendCommand, 186
 setConnectionCallback, 186
 setMessageCallback, 186
NetworkThread, 186
 ~NetworkThread, 187
 isRunning, 187
 NetworkThread, 186
 start, 187
 stop, 187
networkThreadLoop
 NetworkManager, 185
new_connection
 server_run.c, 577
next
 client_s, 89
 egg_s, 100
 graphical_client_s, 114
next_valid_token
 array_function.c, 560
nfds
 server_s, 258
normalize
 raylibcpp::Vector3Wrap, 310

num_agents
main, 46
NUM_RANDOM_MODELS
Constants.hpp, 360

onMapSizeChanged
GameLoop, 108
IGraphicalContextObserver, 123
Zappy::MapRendererObserver, 161

onTileChanged
GameLoop, 108
IGraphicalContextObserver, 123
Zappy::MapRendererObserver, 161

paringArgsClass, 46
paringArgsClass.parseArgs, 188
 __init__, 188
 args, 191
 getFlagValue, 189
 getMachine, 189
 getName, 189
 getPort, 190
 hasFlagAndValue, 190
 machine, 191
 name, 191
 parseArgs, 189
 parsed_args, 191
 port, 191
 searchUnknownFlag, 190
 setMachine, 190
 setName, 190
 setPort, 191

paringArgsClass.ParsingArgsError, 193
 __init__, 193
 message, 194

parse
 ParsingCLI, 195

parse_args
 parsing.c, 580
 parsing.h, 523

parse_client_nb
 parsing.c, 581

parse_command.c
 check_command, 438
 check_disconnect, 439
 execute_com, 439
 execute_graphical_command, 439
 execute_if_free, 439
 find_and_execute, 440
 get_command_data, 440
 get_message, 440
 send_info_new_client, 441

parse_command_utils.c
 add_pending_cmd, 442
 cleanup_client, 442
 cleanup_pending, 443
 cleanup_player_queue, 443
 write_command_output, 443

parse_command_utils_bis.c

 can_connect, 444
 is_valid_team_name, 444

parse_coordinates
 parsing.c, 581

parse_frequence
 parsing.c, 581

parse_inventory
 zappy, 59

parse_names
 parsing.c, 581

parse_port
 parsing.c, 582

parseAllTilesContent
 ProtocolParser, 211

parseArgs
 paringArgsClass.parseArgs, 189

parsed_args
 paringArgsClass.parseArgs, 191

parsed_info
 server_s, 258

parseEggConnection
 ProtocolParser, 212

parseEggDeath
 ProtocolParser, 212

parseEggDrop
 ProtocolParser, 212

parseEggLaying
 ProtocolParser, 212

parseEndGame
 ProtocolParser, 213

ParseFunction
 ProtocolParser, 210

parseIncantationEnd
 ProtocolParser, 213

parseIncantationStart
 ProtocolParser, 213

parseIntParameter
 ProtocolParser, 213

parseMapSize
 ProtocolParser, 213

parseMessage
 ProtocolParser, 214

parsePlayerBroadcast
 ProtocolParser, 214

parsePlayerConnection
 ProtocolParser, 215

parsePlayerDeath
 ProtocolParser, 215

parsePlayerExpulsion
 ProtocolParser, 215

parsePlayerInventory
 ProtocolParser, 215

parsePlayerLevel
 ProtocolParser, 216

parsePlayerPosition
 ProtocolParser, 216

parseRessourceCollect
 ProtocolParser, 216

parseRessourceDrop
 ProtocolParser, 216
parseServerMessage
 ProtocolParser, 217
parseTeamNames
 ProtocolParser, 217
parseTileContent
 ProtocolParser, 217
parseTimeUnit
 ProtocolParser, 217
parseUnknownCommand
 ProtocolParser, 217
parsing.c
 check_args, 579
 check_other_flags, 579
 check_zero, 580
 is_dup_flag, 580
 parse_args, 580
 parse_client_nb, 581
 parse_coordinates, 581
 parse_frequence, 581
 parse_names, 581
 parse_port, 582
parsing.h
 arr_len, 523
 free_arr, 523
 parse_args, 523
 parsing_error, 524
 parsing_info_t, 523
 str_to_word_arr, 524
parsing_err.c
 free_names, 583
 parsing_error, 583
parsing_error
 parsing.h, 524
 parsing_err.c, 583
parsing_info_s, 192
 client_nb, 192
 frequence, 192
 height, 192
 names, 192
 port, 192
 width, 192
parsing_info_t
 parsing.h, 523
ParsingCLI, 194
 ~ParsingCLI, 194
 checkValidityCLI, 195
 displayHelp, 195
 getMachine, 195
 getPort, 195
 parse, 195
 ParsingCLI, 194
 setMachine, 195
 setPort, 196
pause
 raylibcpp::MusicWrap, 181
 raylibcpp::SoundWrap, 268
PBC_HEADER
 HeaderMessage.hpp, 371
PDI_HEADER
 HeaderMessage.hpp, 372
PDR_HEADER
 HeaderMessage.hpp, 372
pending_cmd
 player_s, 199
pending_cmd.h
 client_t, 525
 cmd_func_t, 525
 pending_cmd_t, 525
 server_t, 525
pending_cmd_s, 196
 args, 196
 func, 197
pending_cmd_t
 pending_cmd.h, 525
pending_cmd_utils.c
 execute_pending_cmd, 445
pending_cmd_utils.h
 execute_pending_cmd, 526
pending_requests
 socketManager.SocketManager, 267
PEX_HEADER
 HeaderMessage.hpp, 372
PFK_HEADER
 HeaderMessage.hpp, 372
PGT_HEADER
 HeaderMessage.hpp, 372
PHIRAS
 GraphicalContext.hpp, 362
 tile.h, 542
PHIRAS_D
 tile.h, 541
PIC_HEADER
 HeaderMessage.hpp, 372
pid
 main, 46
PIE_HEADER
 HeaderMessage.hpp, 372
PIN_HEADER
 HeaderMessage.hpp, 372
play
 raylibcpp::MusicWrap, 181
 raylibcpp::SoundWrap, 268
player
 client_s, 89
player.h
 add_item_to_inventory, 528
 DOWN, 528
 free_inventory, 529
 get_inventory_content, 529
 get_resource_name, 529
 how_many_in_inventory, 529
 init_inventory, 530
 init_player, 530
 inventory_has_item, 530

LEFT, 528
player_inventory_t, 528
player_t, 528
remove_item_from_inventory, 530
RIGHT, 528
rotation_e, 528
UP, 528
player_capacity
 tile_s, 294
player_count
 tile_s, 294
player_death.c
 check_player_starvation, 584
 handle_player_death, 584
player_ids
 tile_s, 294
player_init.c
 find_egg_for_team, 586
 init_new_player_pos, 586
 init_pending, 586
 init_player, 586
 init_player_team, 587
player_inventory.c
 add_item_to_inventory, 588
 free_inventory, 588
 init_inventory, 588
 remove_item_from_inventory, 588
player_inventory_display.c
 add_separator, 589
 append_item_string, 589
 get_inventory_content, 590
 get_one_item_content, 590
 process_one_item, 590
player_inventory_s, 197
 name, 197
 quantity, 197
 type, 197
player_inventory_t
 player.h, 528
player_inventory_utils.c
 get_resource_name, 591
 how_many_in_inventory, 592
 inventory_has_item, 592
player_s, 197
 busy_until, 198
 command_queue, 198
 incantation_leader_id, 198
 inventory, 199
 inventory_size, 199
 is_in_incantation, 199
 level, 199
 life, 199
 pending_cmd, 199
 pos_x, 199
 pos_y, 199
 queue_size, 199
 rotation, 199
 team_name, 200
player_t
 player.h, 528
playerIds
 TileData, 299
PlayerInfo
 IMessageData.hpp, 378
PlayerInfoData, 200
 getId, 201
 getLevel, 201
 getOrientation, 201
 getTeamName, 202
 getType, 202
 getX, 202
 getY, 202
 PlayerInfoData, 201
 setId, 202
 setLevel, 202
 setOrientation, 202
 setTeamName, 202
 setX, 203
 setY, 203
PlayerInventory
 IMessageData.hpp, 378
PlayerInventoryData, 203
 getDeraumere, 205
 getFood, 205
 getId, 205
 getLinemate, 205
 getMendiane, 205
 getPhiras, 205
 getSibur, 205
 getThystame, 206
 getType, 206
 getX, 206
 getY, 206
 PlayerInventoryData, 204
 setDeraumere, 206
 setFood, 206
 setId, 206
 setLinemate, 206
 setMendiane, 207
 setPhiras, 207
 setSibur, 207
 setThystame, 207
 setX, 207
 setY, 207
PlayMusic
 IGraphicsLib, 130
 RayLib, 240
PlaySound
 IGraphicsLib, 131
 RayLib, 240
PLV_HEADER
 HeaderMessage.hpp, 372
PNW_HEADER
 HeaderMessage.hpp, 372
poll_client
 server_run.c, 577

poll_manager
 server_s, 258
poll_manager_s, 207
 capacity, 208
 fds, 208
 needs_rebuild, 208
poll_manager_t
 server.h, 532
pollSocket
 SystemWrapper, 55
port
 agent.Agent, 73
 main, 46
 paringArgsClass.parseArgs, 191
 parsing_info_s, 192
pos_x
 egg_s, 100
 player_s, 199
pos_y
 egg_s, 100
 player_s, 199
position
 CameraController, 85
 raylibcpp::RenderState, 246
PPO_HEADER
 HeaderMessage.hpp, 373
print_co
 connection_err.c, 568
 server.h, 537
process_next_queued_command
 command.h, 510
 queue_connection.c, 574
 server.h, 538
process_one_item
 player_inventory_display.c, 590
process_server_message
 agent.Agent, 71
process_tokens
 array_function.c, 560
processIncomingMessages
 NetworkManager, 185
ProtocolParser, 209
 ~ProtocolParser, 210
 extractCommandParameter, 211
 extractMessageParameters, 211
 getCommandName, 211
 handleProtocol, 211
 messageComplete, 211
 parseAllTilesContent, 211
 parseEggConnection, 212
 parseEggDeath, 212
 parseEggDrop, 212
 parseEggLaying, 212
 parseEndGame, 213
 ParseFunction, 210
 parseIncantationEnd, 213
 parseIncantationStart, 213
 parseIntParameter, 213
 parseMapSize, 213
 parseMessage, 214
 parsePlayerBroadcast, 214
 parsePlayerConnection, 215
 parsePlayerDeath, 215
 parsePlayerExpulsion, 215
 parsePlayerInventory, 215
 parsePlayerLevel, 216
 parsePlayerPosition, 216
 parseRessourceCollect, 216
 parseRessourceDrop, 216
 parseServerMessage, 217
 parseTeamNames, 217
 parseTileContent, 217
 parseTimeUnit, 217
 parseUnknownCommand, 217
 ProtocolParser, 210
 splitMessage, 218
push_client
 eject.c, 423
push_eggs
 eject.c, 424
push_single_client
 eject.c, 424
quantity
 player_inventory_s, 197
queue_connection.c
 add_to_command_queue, 573
 check_rebuild, 574
 process_next_queued_command, 574
queue_size
 player_s, 199
r
 ZappyTypes::Color, 90
RayGui, 218
 ~RayGui, 220
 ButtonPressed, 220
 DrawButton, 220
 DrawCheckBox, 220
 DrawColorPicker, 221
 DrawComboBox, 221
 DrawDropdownBox, 221
 DrawGroupBox, 222
 DrawLabel, 222
 DrawListView, 222
 DrawMessageBox, 222
 DrawPanel, 223
 DrawProgressBar, 223
 DrawScrollPane, 223
 DrawSlider, 223
 DrawSpinner, 224
 DrawStatusBar, 224
 DrawTabBar, 224
 DrawTextBox, 225
 DrawTextInputBox, 225
 DrawToggle, 225
 DrawToggleGroup, 225

DrawValueBox, 226
DrawWindowBox, 226
GetStyle, 226
IconText, 226
LoadStyle, 227
RayGui, 220
RayGUICPP::RayGui, 227
SetStyle, 227
RayGui.cpp
 createGuiLib, 317
raygui_impl.cpp
 RAYGUI_IMPLEMENTATION, 319
 RAYLIB_COLOR_DEFINED, 319
 RAYLIB_RECTANGLE_DEFINED, 319
 RAYLIB_VECTOR2_DEFINED, 319
 RAYLIB_VECTOR3_DEFINED, 320
RAYGUI_IMPLEMENTATION
 raygui_impl.cpp, 319
RayGUICPP, 46
RayGUICPP::Button, 79
 Draw, 80
RayGUICPP::CheckBox, 87
 Draw, 87
RayGUICPP::ColorPicker, 90
 Draw, 90
RayGUICPP::ComboBox, 91
 Draw, 91
RayGUICPP::DropdownBox, 97
 Draw, 98
RayGUICPP::GroupBox, 119
 Draw, 119
RayGUICPP::Icon, 120
 IconText, 120
RayGUICPP::Label, 148
 Draw, 149
RayGUICPP::ListView, 152
 Draw, 152
RayGUICPP::MessageBox, 166
 Draw, 166
RayGUICPP::Panel, 187
 Draw, 188
RayGUICPP::ProgressBar, 208
 Draw, 209
RayGUICPP::RayGui, 227
 ~RayGui, 228
 GetStyle, 228
 LoadStyle, 228
 RayGui, 227
 SetStyle, 228
RayGUICPP::ScrollPane, 255
 Draw, 255
RayGUICPP::Slider, 264
 Draw, 264
RayGUICPP::Spinner, 269
 Draw, 269
RayGUICPP::StatusBar, 269
 Draw, 270
RayGUICPP::TabBar, 270
 Draw, 270
RayGUICPP::TextBox, 284
 Draw, 285
RayGUICPP::TextInputBox, 285
 Draw, 285
RayGUICPP::Toggle, 302
 Draw, 302
RayGUICPP::ToggleGroup, 302
 Draw, 303
RayGUICPP::ValueBox, 307
 Draw, 307
RayGUICPP::WindowBox, 313
 Draw, 313
RayLib, 228
 ~RayLib, 230
 BeginCamera3D, 231
 BeginDrawing, 231
 BindTexture3D, 231
 ClearBackground, 231
 CloseWindow, 231
 CreateCamera2D, 232
 CreateCamera3D, 232
 DrawCircle, 232
 DrawCube, 232
 DrawCylinder, 233
 DrawGrid, 233
 DrawLine3D, 233
 DrawModel3D, 233
 DrawModelEx, 234
 DrawPlane, 234
 DrawRectangle, 234
 DrawSphere, 234
 DrawText, 235
 DrawText3D, 235
 DrawTexture2D, 235
 DrawTextWave3D, 235
 EndCamera3D, 236
 EndDrawing, 236
 GetMousePosition, 236
 GetMouseWheelMove, 236
 GetMouseX, 237
 GetMouseY, 237
 InitWindow, 237
 IsKeyDown, 237
 IsKeyPressed, 237
 IsMouseButtonDown, 237
 IsMouseButtonPressed, 238
 IsMouseButtonReleased, 238
 IsTextureReady, 238
 LoadFont, 238
 LoadModel3D, 238
 LoadModelWithTexture, 239
 LoadTexture2D, 239
 LoadTexture3D, 239
 MeasureText3D, 239
 PlayMusic, 240
 PlaySound, 240
 RayLib, 230

setFPS, 240
 SetMusicVolume, 240
 SetSoundVolume, 240
 StopMusic, 241
 StopSound, 241
 UnloadFont, 241
 UnloadModel3D, 241
 UnloadTexture2D, 241
 UnloadTexture3D, 241
 UpdateMusic, 242
 WindowShouldClose, 242
RayLib.cpp
 createGraphicsLib, 343
RAYLIB_COLOR_DEFINED
 raygui_impl.cpp, 319
RAYLIB_RECTANGLE_DEFINED
 raygui_impl.cpp, 319
RAYLIB_VECTOR2_DEFINED
 raygui_impl.cpp, 319
RAYLIB_VECTOR3_DEFINED
 raygui_impl.cpp, 320
raylibcpp, 47
 RenderStatePtr, 48
 WaveTextConfigPtr, 48
 WaveTextConfigUPtr, 48
raylibcpp::Audio, 74
 ~Audio, 74
 Audio, 74
raylibcpp::Camera2DWrap, 80
 ~Camera2DWrap, 80
 Camera2DWrap, 80
 get, 80
raylibcpp::Camera3DWrap, 81
 ~Camera3DWrap, 81
 beginMode, 82
 Camera3DWrap, 81
 endMode, 82
 get, 82
raylibcpp::Font, 105
 ~Font, 106
 drawText, 106
 Font, 105
 get, 106
raylibcpp::GlyphDimensions, 113
 height, 113
 width, 113
raylibcpp::Input, 146
 getMousePosition, 146
 getMouseWheelMove, 146
 getX, 146
 getY, 146
 isKeyDown, 146
 isKeyPressed, 147
 isMouseButtonDown, 147
 isMouseButtonPressed, 147
 isMouseButtonReleased, 147
raylibcpp::ModelWrap, 179
 ~ModelWrap, 179
 draw, 180
 get, 180
 ModelWrap, 179
raylibcpp::MusicWrap, 180
 ~MusicWrap, 181
 get, 181
 MusicWrap, 180
 pause, 181
 play, 181
 resume, 181
 setVolume, 181
 stop, 182
 update, 182
raylibcpp::RenderState, 246
 charIndex, 246
 position, 246
 textOffsetX, 246
 textOffsetY, 246
raylibcpp::Shape, 260
 drawCircle, 260
 drawCube, 261
 drawCylinder, 261
 drawGrid, 261
 drawLine3D, 261
 drawPlane, 261
 drawRectangle, 262
 drawSphere, 262
raylibcpp::SoundWrap, 267
 ~SoundWrap, 268
 get, 268
 pause, 268
 play, 268
 resume, 268
 setVolume, 268
 SoundWrap, 268
 stop, 269
raylibcpp::Text3D, 274
 ~Text3D, 275
 DrawText3D, 275
 DrawTextCodepoint3D, 275
 DrawTextWave3D, 275
 MeasureTextWave3D, 276
 Text3D, 275
raylibcpp::Text3DCodepoint, 276
 draw, 276
raylibcpp::Text3DConfig, 277
 adjustWaveIntensity, 277
 adjustWaveSpeed, 277
 createDefaultWaveConfig, 278
 createFastWave, 278
 createIntenseWave, 278
 createSlowWave, 278
 createSubtleWave, 278
 createWaveConfig, 279
 isShowLetterBoundary, 279
 isShowTextBoundary, 279
 isValidWaveConfig, 279
 setShowLetterBoundary, 279

setShowTextBoundary, 280
raylibcpp::Text3DHelper, 48
calculateAdvanceX, 48
calculateGlyphDimensions, 49
calculateGlyphPosition, 49
calculateScale, 49
calculateSourceRectangle, 49
calculateTextureCoordinates, 50
getGlyphIndex, 50
isNewline, 50
isRenderableCharacter, 50
isValidFont, 50
isValidRenderParams, 50
isWhitespace, 51
MAX_FONT_SIZE, 51
MAX_SPACING, 51
MIN_FONT_SIZE, 51
MIN_SPACING, 51
raylibcpp::Text3DMeasurement, 280
measureText, 280
raylibcpp::Text3DParser, 281
isWaveMarker, 281
renderWaveText, 281
skipWaveMarkers, 282
raylibcpp::Text3DRenderer, 282
renderBasicText, 283
raylibcpp::Text3DWave, 283
applyWaveEffect, 284
createDefaultConfig, 284
isValidConfig, 284
raylibcpp::Texture, 286
~Texture, 286
draw, 286
get, 286
getNextId, 287
isReady, 287
Texture, 286
raylibcpp::Texture3DWrap, 287
~Texture3DWrap, 287
bind, 288
get, 288
Texture3DWrap, 287
raylibcpp::TextureCoordinates, 288
th, 288
tw, 288
tx, 289
ty, 289
raylibcpp::Utils, 306
hexToColor, 307
raylibcpp::Vector3Wrap, 309
add, 309
cross, 309
get, 310
normalize, 310
scale, 310
set, 310
sub, 310
transform, 310
Vector3Wrap, 309
raylibcpp::WaveTextConfig, 311
waveOffset, 311
waveRange, 311
waveSpeed, 311
raylibcpp::Window, 311
~Window, 312
beginDrawing, 312
clear, 312
endDrawing, 312
setFps, 313
shouldClose, 313
Window, 312
readSocket
 SystemWrapper, 55
receive
 TcpConnection, 272
remove_egg
 egg.c, 549
 egg.h, 515
remove_egg_by_id
 egg.c, 550
remove_egg_from_tile
 egg.c, 550
 tile.h, 543
remove_fd
 connection.c, 566
 server.h, 538
remove_graphic_client
 client.h, 494
 graphical_client.c, 572
remove_graphical_node
 graphical_client.c, 572
remove_head_client
 connection.c, 566
remove_id_from_array
 egg.c, 550
remove_item_from_inventory
 player.h, 530
 player_inventory.c, 588
remove_node
 egg.c, 550
remove_other_client
 connection.c, 567
remove_resources
 finish_incantation.c, 427
removeObserver
 GraphicalContext, 118
render
 Renderer, 244
 Zappy::MapRenderer, 157
renderBackground
 Renderer, 245
renderBasicText
 raylibcpp::Text3DRenderer, 283
renderCameraInfo
 UIRenderer, 304
renderCube

GameLoop, 108
 Renderer, 243
 ~Renderer, 243
 getMapHeight, 244
 getMapWidth, 244
 getResourceTextureId, 244
 init, 244
 loadResourceTexture, 244
 render, 244
 renderBackground, 245
 Renderer, 243
 renderGrid, 245
 renderModelFromManager, 245
 renderScene, 245
 renderSprite2D, 245
 renderUI, 246
 renderFPS
 UIRenderer, 304
 renderGrid
 Renderer, 245
 renderModelFromManager
 Renderer, 245
 renderScene
 Renderer, 245
 renderSprite2D
 Renderer, 245
 RenderStatePtr
 raylibcpp, 48
 renderStatusBar
 UIRenderer, 304
 renderTile
 Zappy::DetailedTileRenderStrategy, 95
 Zappy::ITileRenderStrategy, 148
 Zappy::ModelTileRenderStrategy, 178
 Zappy::SimpleTileRenderStrategy, 263
 renderUI
 Renderer, 246
 UIRenderer, 304
 renderWaveText
 raylibcpp::Text3DParser, 281
 repeat_word
 tile_to_str.c, 455
 reset
 CameraController, 85
 resize
 SystemWrapper::SafeBuffer, 250
 Resource
 IMessageData.hpp, 378
 resource_dist_s, 247
 height, 247
 map, 247
 tile_indices, 247
 width, 248
 resource_dist_t
 tile.h, 541
 resource_type
 tile.h, 542
 resource_type_t

tile.h, 541
 ResourceData, 248
 getPlayerId, 249
 getResourceType, 249
 getType, 249
 ResourceData, 249
 setPlayerId, 249
 setResourceType, 249
 resources, 51
 resources, 51
 tile_s, 294
 TileData, 299
 resources_map.c
 distribute_one_resource, 552
 distribute_resources, 553
 init_resource_dist, 553
 respawn_resources, 553
 shuffle_indices, 554
 ResourceType
 GraphicalContext.hpp, 362
 respawn_resources
 resources_map.c, 553
 tile.h, 543
 resume
 raylibcpp::MusicWrap, 181
 raylibcpp::SoundWrap, 268
 RIGHT
 player.h, 528
 right
 command.h, 510
 right.c, 447
 right.c
 change_rot, 446
 right, 447
 roles, 52
 roles, 52
 rotation
 player_s, 199
 rotation_e
 player.h, 528
 run
 agent.Agent, 71
 GameLoop, 108
 run_agent
 main, 45
 running
 socketManager.SocketManager, 267
 s_fd
 server_s, 258
 SafeBuffer
 SystemWrapper::SafeBuffer, 250
 SafePollFd
 SystemWrapper::SafePollFd, 252
 SafeSockAddr
 SystemWrapper::SafeSockAddr, 254
 SBP_HEADER
 HeaderMessage.hpp, 373
 scale

Model3D, 169
raylibcpp::Vector3Wrap, 310
searchUnknownFlag
 paringArgsClass.parseArgs, 190
secret_key
 encryption, 44
SEG_HEADER
 HeaderMessage.hpp, 373
send
 TcpConnection, 272
send_all_player_info_to_one_client
 command_pnw.c, 481
 graphical_commands.h, 518
send_bct_command
 command_bct.c, 460
send_bct_to_all_graphical_clients
 command.h, 510
 command_bct.c, 461
send_broadcast_to_client
 broadcast.c, 419
send_command
 agent.Agent, 71
 socketManager.SocketManager, 265
send_ebo_command
 command_ebo.c, 462
 graphical_commands.h, 518
send_edi_command
 command_edi.c, 463
 graphical_commands.h, 519
send_enw_command
 command.h, 510
 command_enw.c, 465
send_enw_command_start
 command.h, 511
 command_enw.c, 465
send_enw_command_to_client
 command_enw.c, 465
send_info_new_client
 parse_command.c, 441
send_map_info_to_one_client
 client.h, 494
 graphical_client.c, 572
send_message_to_all_graphic
 client.h, 495
 graphical_client.c, 572
send_msz_command
 command_msz.c, 466
 graphical_commands.h, 519
send_mtc_to_all_graphical_clients
 command.h, 511
 command_bct.c, 461
send_one_tna_command
 command_tna.c, 489
send_pin_command
 command_pin.c, 477
 graphical_commands.h, 519
send_pin_to_all
 command_pin.c, 478
 graphical_commands.h, 519
send_plv_command
 command_plv.c, 479
 graphical_commands.h, 520
send_plv_to_all
 command_plv.c, 479
 graphical_commands.h, 520
send_pnw_command
 command_pnw.c, 481
 graphical_commands.h, 520
send_pnw_command_to_all
 command.h, 511
 command_pnw.c, 481
send_ppo_command
 command_ppo.c, 483
 graphical_commands.h, 521
send_smg_command
 command_smg.c, 486
 graphical_commands.h, 521
send_team_names_to_one_client
 graphical_commands.h, 521
send_tile_content_to_one_client
 command_bct.c, 461
 graphical_commands.h, 522
send_tna_command
 command_tna.c, 489
 graphical_commands.h, 522
sendCommand
 NetworkManager, 186
serv_add
 server_s, 258
server.h
 add_fd, 533
 add_to_command_queue, 533
 check_client, 533
 count_team, 534
 create_map, 534
 create_server, 535
 display_help, 535
 finish_incantation, 535
 free_all, 536
 free_node, 536
 init_new_player_pos, 537
 init_server_eggs, 537
 poll_manager_t, 532
 print_co, 537
 process_next_queued_command, 538
 remove_fd, 538
 server_err, 539
 server_t, 532
 update_game_tick, 539
server_err
 connection_err.c, 568
 server.h, 539
server_loop
 main.c, 546
server_run.c
 check_client, 575

check_client_message, 576
 check_new_connection, 576
 fill_poll_array, 576
 find_client_by_socket, 576
 handle_game_tick, 577
 new_connection, 577
 poll_client, 577
 setup_poll_manager, 578
 smart_polling, 578
server_s, 256
 client, 257
 current_resources, 257
 current_tick, 257
 eggs, 257
 graphical_clients, 257
 map, 257
 nfds, 258
 parsed_info, 258
 poll_manager, 258
 s_fd, 258
 serv_add, 258
 should_run, 258
 total_resources, 258
server_t
 egg.h, 514
 pending_cmd.h, 525
 server.h, 532
 tile.h, 541
ServerMessage
 IMessageData.hpp, 378
ServerMessageData, 259
 getMessage, 260
 getType, 260
 ServerMessageData, 259
 setMessage, 260
set
 raylibcpp::Vector3Wrap, 310
set_busy_all
 start_incantation.c, 451
set_busy_until
 start_incantation.c, 451
set_object
 command.h, 512
 set_object.c, 448
set_object.c
 set_object, 448
set_rdm_seed
 main.c, 547
setConnectionCallback
 NetworkManager, 186
setData
 Message, 165
 SystemWrapper::SafeBuffer, 251
setDeraumere
 PlayerInventoryData, 206
 TileContentData, 297
setDetailThreshold
 Zappy::MapRenderer, 158
setDragging
 CameraController, 85
setEggId
 EggData, 103
setEnabled
 NetworkLogger, 183
setEvents
 SystemWrapper::SafePollFd, 253
setFd
 SystemWrapper::SafePollFd, 253
setFood
 PlayerInventoryData, 206
 TileContentData, 297
setFps
 IGraphicsLib, 131
 RayLib, 240
 raylibcpp::Window, 313
setGraphicsLib
 ModelManager, 175
 TextureManager, 292
 Zappy::ModelManagerAdapter, 177
setHeader
 Message, 166
setHeight
 MapSizeData, 163
setId
 PlayerInfoData, 202
 PlayerInventoryData, 206
setLevel
 IncantationData, 143
 PlayerInfoData, 202
setLinemate
 PlayerInventoryData, 206
 TileContentData, 297
setMachine
 paringArgsClass.parseArgs, 190
 ParsingCLI, 195
setMapDimensions
 CameraController, 85
setMendiane
 PlayerInventoryData, 207
 TileContentData, 297
setMessage
 BroadcastData, 78
 Message, 166
 ServerMessageData, 260
setMessageCallback
 NetworkManager, 186
setMousePosition
 CameraController, 86
SetMusicVolume
 IGraphicsLib, 131
 RayLib, 240
setName
 paringArgsClass.parseArgs, 190
setNames
 TeamNameData, 274
setNonBlocking

SystemWrapper, 55
setOrientation
 PlayerInfoData, 202
setPhiras
 PlayerInventoryData, 207
 TileContentData, 297
setPlayerId
 BroadcastData, 78
 EggData, 103
 ResourceData, 249
setPlayerIds
 IncantationData, 143
setPort
 paringArgsClass.parseArgs, 191
 ParsingCLI, 196
setResourceColor
 Zappy::MapRenderer, 158
setResourceRenderStrategy
 Zappy::MapRenderer, 158
setResourceType
 ResourceData, 249
setRevents
 SystemWrapper::SafePollFd, 253
setServerInfo
 GameLoop, 109
setShowLetterBoundary
 raylibcpp::Text3DConfig, 279
setShowTextBoundary
 raylibcpp::Text3DConfig, 280
setSibur
 PlayerInventoryData, 207
 TileContentData, 297
SetSoundVolume
 IGraphicsLib, 131
 RayLib, 240
SetStyle
 IGuiLib, 138
 RayGui, 227
 RayGUICPP::RayGui, 228
setSuccess
 IncantationEndData, 145
setTeamName
 EndGameData, 105
 PlayerInfoData, 202
setThystame
 PlayerInventoryData, 207
 TileContentData, 298
setTileRenderStrategy
 Zappy::MapRenderer, 158
setSize
 Zappy::MapRenderer, 159
setTileSpacing
 Zappy::MapRenderer, 159
setTimeUnit
 TimeUnitData, 302
setup_poll_manager
 server_run.c, 578
setup_signal_handler
 main.c, 547
setVolume
 raylibcpp::MusicWrap, 181
 raylibcpp::SoundWrap, 268
setWidth
 MapSizeData, 163
setX
 EggData, 103
 IncantationData, 143
 IncantationEndData, 145
 PlayerInfoData, 203
 PlayerInventoryData, 207
 TileContentData, 298
setY
 EggData, 103
 IncantationData, 143
 IncantationEndData, 145
 PlayerInfoData, 203
 PlayerInventoryData, 207
 TileContentData, 298
setZoomLevel
 Zappy::MapRenderer, 159
SGT_HEADER
 HeaderMessage.hpp, 373
should_continue_running
 main.c, 547
should_run
 server_s, 258
shouldClose
 raylibcpp::Window, 313
showUI
 UIRenderer, 305
shuffle_indices
 resources_map.c, 554
SIBUR
 GraphicalContext.hpp, 362
 tile.h, 542
SIBUR_D
 tile.h, 541
sigint_handler
 main.c, 547
signal_handler
 main, 45
sigpipe_handler
 main.c, 548
SimpleTileRenderStrategy
 Zappy::SimpleTileRenderStrategy, 263
size
 MessageQueue, 168
 SystemWrapper::SafeBuffer, 251
skipWaveMarkers
 raylibcpp::Text3DParser, 282
smart_polling
 server_run.c, 578
SMG_HEADER
 HeaderMessage.hpp, 373
sock
 agent.Agent, 73

socketManager.SocketManager, 267
 socketManager, 52
 agent.Agent, 73
 socketManager.SocketManager, 264
 __init__, 265
 buffer, 266
 get_message, 265
 has_messages, 265
 lock, 266
 message_queue, 267
 pending_requests, 267
 running, 267
 send_command, 265
 sock, 267
 start, 266
 stop, 266
 thread, 267
 SoundWrap
 raylibcpp::SoundWrap, 268
 splitMessage
 ProtocolParser, 218
 SST_HEADER
 HeaderMessage.hpp, 373
 start
 agent.Agent, 72
 circular_buffer_s, 88
 NetworkThread, 187
 socketManager.SocketManager, 266
 start_incantation
 command.h, 512
 start_incantation.c, 451
 start_incantation.c
 can_start_incantation, 449
 get_client_by_id_player, 449
 has_enough_resources, 450
 how_many_players_needed, 450
 nb_valid_players_on_tile, 450
 set_busy_all, 451
 set_busy_until, 451
 start_incantation, 451
 stop
 agent.Agent, 72
 NetworkThread, 187
 raylibcpp::MusicWrap, 182
 raylibcpp::SoundWrap, 269
 socketManager.SocketManager, 266
 StopMusic
 IGraphicsLib, 131
 RayLib, 241
 StopSound
 IGraphicsLib, 131
 RayLib, 241
 str_to_word_arr
 array_function.c, 561
 command.h, 512
 parsing.h, 524
 sub
 raylibcpp::Vector3Wrap, 310
 SUC_HEADER
 HeaderMessage.hpp, 373
 SystemWrapper, 52
 closeSocket, 53
 connectSocket, 53
 createSocket, 53
 executeWithTimeout, 53
 getErrorString, 54
 getSocketOption, 54
 inetNtop, 54
 inetPton, 54
 makeSharedArray, 54
 makeUniqueArray, 54
 pollSocket, 55
 readSocket, 55
 setNonBlocking, 55
 writeSocket, 56
 SystemWrapper::SafeBuffer, 250
 ~SafeBuffer, 250
 data, 250
 resize, 250
 SafeBuffer, 250
 setData, 251
 size, 251
 toString, 251
 SystemWrapper::SafePollFd, 251
 getEvents, 252
 getFd, 252
 getPollFd, 252
 getPollPtr, 252, 253
 getRevents, 253
 SafePollFd, 252
 setEvents, 253
 setFd, 253
 setRevents, 253
 SystemWrapper::SafeSockAddr, 253
 getAddr, 254
 getSize, 254
 getSockAddr, 255
 SafeSockAddr, 254
 take_action
 decisionManager.DecisionManager, 93
 take_all_of_item_here
 actions, 42
 take_everything_here
 actions, 42
 take_object
 command.h, 513
 take_object.c, 452
 take_object.c
 determine_type, 452
 take_object, 452
 update_resources, 453
 target
 CameraController, 86
 TcpConnection, 270
 ~TcpConnection, 271
 close, 271

connect, 271
hasData, 272
isConnected, 272
receive, 272
send, 272
 TcpConnection, 271

team
 agent.Agent, 73
 main, 46

team_name
 egg_s, 100
 player_s, 200

TeamName
 IMessageData.hpp, 378

TeamNameData, 273
 addName, 274
 getNames, 274
 getType, 274
 setNames, 274
 TeamNameData, 273

Text3D
 raylibcpp::Text3D, 275

Text3DCore.hpp
 LETTER_BOUNDARY_COLOR, 331
 LETTER_BOUNDARY_SIZE, 331
 TEXT_MAX_LAYERS, 332

TEXT_MAX_LAYERS
 Text3DCore.hpp, 332

textOffsetX
 raylibcpp::RenderState, 246

textOffsetY
 raylibcpp::RenderState, 246

Texture
 raylibcpp::Texture, 286

Texture3DWrap
 raylibcpp::Texture3DWrap, 287

textureIds
 Model3D, 170

TextureManager, 289
 drawTexture, 290
 getInstance, 290
 getTextureCount, 290
 getTextureId, 290
 hasTexture, 291
 hasTextureLibSet, 291
 loadTexture, 291
 setGraphicsLib, 292
 unloadAllTextures, 292
 unloadTexture, 292

th
 raylibcpp::TextureCoordinates, 288

thread
 socketManager.SocketManager, 267

THYSTAME
 GraphicalContext.hpp, 362
 tile.h, 542

THYSTAME_D
 tile.h, 541

tick_check
 tick_update.c, 555

tick_update.c
 tick_check, 555
 update_game_tick, 555

tile.c
 tile_add_player, 556
 tile_free, 557
 tile_init, 557
 tile_remove_player, 557

tile.h
 add_egg_to_tile, 542
 COUNT, 542
 DERAUMERE, 542
 DERAUMERE_D, 540
 distribute_resources, 542
 FOOD, 542
 FOOD_D, 540
 LINEMATE, 542
 LINEMATE_D, 541
 MENDIANE, 542
 MENDIANE_D, 541
 PHIRAS, 542
 PHIRAS_D, 541
 remove_egg_from_tile, 543
 resource_dist_t, 541
 resource_type, 542
 resource_type_t, 541
 respawn_resources, 543
 server_t, 541
 SIBUR, 542
 SIBUR_D, 541
 THYSTAME, 542
 THYSTAME_D, 541
 tile_add_player, 543
 tile_free, 544
 tile_init, 544
 tile_remove_player, 544
 tile_t, 541

tile_add_player
 tile.c, 556
 tile.h, 543

tile_free
 tile.c, 557
 tile.h, 544

tile_indices
 resource_dist_s, 247

tile_init
 tile.c, 557
 tile.h, 544

tile_remove_player
 tile.c, 557
 tile.h, 544

tile_s, 293
 egg_capacity, 293
 egg_count, 293
 egg_ids, 293
 player_capacity, 294

player_count, 294
 player_ids, 294
 resources, 294
tile_t
 egg.h, 514
 tile.h, 541
tile_to_str
 command.h, 513
 tile_to_str.c, 455
tile_to_str.c
 add_resources, 454
 calculate_total_size, 454
 format_str_resources, 454
 repeat_word, 455
 tile_to_str, 455
TileContent
 IMessageData.hpp, 378
TileContentData, 294
 getDeraumere, 296
 getFood, 296
 getLinemate, 296
 getMendiane, 296
 getPhiras, 296
 getSibur, 296
 getThystame, 296
 getType, 296
 getX, 297
 getY, 297
 setDeraumere, 297
 setFood, 297
 setLinemate, 297
 setMendiane, 297
 setPhiras, 297
 setSibur, 297
 setThystame, 298
 setX, 298
 setY, 298
TileContentData, 295
TileData, 298
 eggIds, 299
 isIncantating, 299
 playerIds, 299
 resources, 299
TileRenderStrategyFactory
 Zappy::TileRenderStrategyFactory, 300
times
 command_data_s, 92
TimeUnit
 IMessageData.hpp, 378
TimeUnitData, 300
 getTimeUnit, 301
 getType, 302
 setTimeUnit, 302
 TimeUnitData, 301
TNA_HEADER
 HeaderMessage.hpp, 373
ToRaylib
 TypeAdapter, 57
toString
 SystemWrapper::SafeBuffer, 251
total_resources
 server_s, 258
transform
 raylibcpp::Vector3Wrap, 310
tw
 raylibcpp::TextureCoordinates, 288
tx
 raylibcpp::TextureCoordinates, 289
ty
 raylibcpp::TextureCoordinates, 289
type
 client_s, 89
 player_inventory_s, 197
TypeAdapter, 56
 FromRaylib, 56, 57
 ToRaylib, 57
UIRenderer, 303
 ~UIRenderer, 303
 fps, 304
 renderCameralInfo, 304
 renderFPS, 304
 renderStatusBar, 304
 renderUI, 304
 showUI, 305
 UIRenderer, 303
unloadAllModels
 ModelManager, 175
unloadAllTextures
 TextureManager, 292
UnloadFont
 IGraphicsLib, 131
 RayLib, 241
unloadModel
 ModelManager, 175
UnloadModel3D
 IGraphicsLib, 131
 RayLib, 241
unloadTexture
 TextureManager, 292
UnloadTexture2D
 IGraphicsLib, 131
 RayLib, 241
UnloadTexture3D
 IGraphicsLib, 132
 RayLib, 241
UP
 player.h, 528
up
 CameraController, 86
update
 CameraController, 86
 raylibcpp::MusicWrap, 182
update_game_tick
 server.h, 539
 tick_update.c, 555
update_resources

take_object.c, 453
updateCameraPosition
 CameraController, 86
updateContext
 GraphicalContext, 119
UpdateFunction
 GraphicalContext, 115
UpdateMusic
 IGraphicsLib, 132
 RayLib, 242
upgrades, 58

Vector3Wrap
 raylibcpp::Vector3Wrap, 309

WARNING
 logger.LogLevel, 156
warning
 logger.Logger, 154
waveOffset
 raylibcpp::WaveTextConfig, 311
waveRange
 raylibcpp::WaveTextConfig, 311
waveSpeed
 raylibcpp::WaveTextConfig, 311
WaveTextConfigPtr
 raylibcpp, 48
WaveTextConfigUPtr
 raylibcpp, 48
what
 AException, 69
 IException, 122
width
 parsing_info_s, 192
 raylibcpp::GlyphDimensions, 113
 resource_dist_s, 248
 ZappyTypes::Rectangle, 242
Window
 raylibcpp::Window, 312
WindowShouldClose
 IGraphicsLib, 132
 RayLib, 242
wrap_position
 eject.c, 424
 forward.c, 430
write_command_output
 command.h, 513
 parse_command_utils.c, 443
writeSocket
 SystemWrapper, 56

x
 ZappyTypes::Rectangle, 242
 ZappyTypes::Vector2, 308
 ZappyTypes::Vector3, 308

y
 ZappyTypes::Rectangle, 243
 ZappyTypes::Vector2, 308

ZappyTypes::Vector3, 308

z
 ZappyTypes::Vector3, 308
Z_BEIGE
 ZappyTypes::Colors, 63
Z_BLACK
 ZappyTypes::Colors, 63
Z_BLANK
 ZappyTypes::Colors, 63
Z_BLUE
 ZappyTypes::Colors, 63
Z_BROWN
 ZappyTypes::Colors, 63
Z_DARKBLUE
 ZappyTypes::Colors, 63
Z_DARKBROWN
 ZappyTypes::Colors, 64
Z_DARKGRAY
 ZappyTypes::Colors, 64
Z_DARKGREEN
 ZappyTypes::Colors, 64
Z_DARKPURPLE
 ZappyTypes::Colors, 64
Z_GOLD
 ZappyTypes::Colors, 64
Z_GRAY
 ZappyTypes::Colors, 64
Z_GREEN
 ZappyTypes::Colors, 64
Z_KEY_A
 ZappyTypes, 61
Z_KEY_APOSTROPHE
 ZappyTypes, 61
Z_KEY_B
 ZappyTypes, 61
Z_KEY_C
 ZappyTypes, 61
Z_KEY_COMMA
 ZappyTypes, 61
Z_KEY_D
 ZappyTypes, 61
Z_KEY_E
 ZappyTypes, 61
Z_KEY_EIGHT
 ZappyTypes, 61
Z_KEY_ENTER
 ZappyTypes, 62
Z_KEY_EQUAL
 ZappyTypes, 61
Z_KEY_ESCAPE
 ZappyTypes, 62
Z_KEY_F
 ZappyTypes, 61
Z_KEY_FIVE
 ZappyTypes, 61
Z_KEY_FOUR
 ZappyTypes, 61
Z_KEY_G

ZappyTypes, 61
Z_KEY_H
 ZappyTypes, 61
Z_KEY_I
 ZappyTypes, 61
Z_KEY_J
 ZappyTypes, 61
Z_KEY_K
 ZappyTypes, 61
Z_KEY_L
 ZappyTypes, 61
Z_KEY_M
 ZappyTypes, 61
Z_KEY_MINUS
 ZappyTypes, 61
Z_KEY_N
 ZappyTypes, 61
Z_KEY_NINE
 ZappyTypes, 61
Z_KEY_NULL
 ZappyTypes, 61
Z_KEY_O
 ZappyTypes, 61
Z_KEY_ONE
 ZappyTypes, 61
Z_KEY_P
 ZappyTypes, 61
Z_KEY_PERIOD
 ZappyTypes, 61
Z_KEY_Q
 ZappyTypes, 61
Z_KEY_R
 ZappyTypes, 61
Z_KEY_S
 ZappyTypes, 61
Z_KEY_SEMICOLON
 ZappyTypes, 61
Z_KEY_SEVEN
 ZappyTypes, 61
Z_KEY_SIX
 ZappyTypes, 61
Z_KEY_SLASH
 ZappyTypes, 61
Z_KEY_SPACE
 ZappyTypes, 62
Z_KEY_T
 ZappyTypes, 61
Z_KEY_THREE
 ZappyTypes, 61
Z_KEY_TWO
 ZappyTypes, 61
Z_KEY_U
 ZappyTypes, 62
Z_KEY_V
 ZappyTypes, 62
Z_KEY_W
 ZappyTypes, 62
Z_KEY_X
 ZappyTypes, 62
ZappyTypes, 62
Z_KEY_Y
 ZappyTypes, 62
Z_KEY_Z
 ZappyTypes, 62
Z_KEY_ZERO
 ZappyTypes, 61
Z_LIGHTGRAY
 ZappyTypes::Colors, 64
Z_LIME
 ZappyTypes::Colors, 65
Z_MAGENTA
 ZappyTypes::Colors, 65
Z_MAROON
 ZappyTypes::Colors, 65
Z_ORANGE
 ZappyTypes::Colors, 65
Z_PINK
 ZappyTypes::Colors, 65
Z_PURPLE
 ZappyTypes::Colors, 65
Z_RAYWHITE
 ZappyTypes::Colors, 65
Z_RED
 ZappyTypes::Colors, 65
Z_SKYBLUE
 ZappyTypes::Colors, 65
Z_VIOLET
 ZappyTypes::Colors, 66
Z_WHITE
 ZappyTypes::Colors, 66
Z_YELLOW
 ZappyTypes::Colors, 66
Zappy, 60
zappy, 58
 get_best_available_resource, 58
 get_closest_of_item, 58
 get_item_relative_pos, 59
 go_get_item, 59
 how_much_of_item_here, 59
 parse_inventory, 59
Zappy::DetailedTileRenderStrategy, 93
 DetailedTileRenderStrategy, 94
 renderTile, 95
Zappy::ITileRenderStrategy, 147
 ~ITileRenderStrategy, 148
 renderTile, 148
Zappy::MapRenderer, 156
 initialize, 157
 MapRenderer, 157
 render, 157
 setDetailThreshold, 158
 setResourceColor, 158
 setResourceRenderStrategy, 158
 setTileRenderStrategy, 158
 setTileSize, 159
 setTileSpacing, 159
 setZoomLevel, 159

Zappy::MapRendererObserver, 160
MapRendererObserver, 160
onMapSizeChanged, 161
onTileChanged, 161
Zappy::ModelManagerAdapter, 176
createShared, 176
getManager, 176
ModelManagerAdapter, 176
setGraphicsLib, 177
Zappy::ModelTileRenderStrategy, 177
ModelTileRenderStrategy, 178
renderTile, 178
Zappy::SimpleTileRenderStrategy, 262
renderTile, 263
SimpleTileRenderStrategy, 263
Zappy::TileRenderStrategyFactory, 299
createDetailedTileStrategy, 300
createModelTileStrategy, 300
createSimpleTileStrategy, 300
TileRenderStrategyFactory, 300
ZappyTypes, 60
KeyboardKey, 61
Z_KEY_A, 61
Z_KEY_APOSTROPHE, 61
Z_KEY_B, 61
Z_KEY_C, 61
Z_KEY_COMMAS, 61
Z_KEY_D, 61
Z_KEY_E, 61
Z_KEY_EIGHT, 61
Z_KEY_ENTER, 62
Z_KEY_EQUAL, 61
Z_KEY_ESCAPE, 62
Z_KEY_F, 61
Z_KEY_FIVE, 61
Z_KEY_FOUR, 61
Z_KEY_G, 61
Z_KEY_H, 61
Z_KEY_I, 61
Z_KEY_J, 61
Z_KEY_K, 61
Z_KEY_L, 61
Z_KEY_M, 61
Z_KEY_MINUS, 61
Z_KEY_N, 61
Z_KEY_NINE, 61
Z_KEY_NULL, 61
Z_KEY_O, 61
Z_KEY_ONE, 61
Z_KEY_P, 61
Z_KEY_PERIOD, 61
Z_KEY_Q, 61
Z_KEY_R, 61
Z_KEY_S, 61
Z_KEY_SEMICOLON, 61
Z_KEY_SEVEN, 61
Z_KEY_SIX, 61
Z_KEY_SLASH, 61
Z_KEY_SPACE, 62
Z_KEY_T, 61
Z_KEY_THREE, 61
Z_KEY_TWO, 61
Z_KEY_U, 62
Z_KEY_V, 62
Z_KEY_W, 62
Z_KEY_X, 62
Z_KEY_Y, 62
Z_KEY_Z, 62
Z_KEY_ZERO, 61
ZappyTypes::Color, 89
a, 90
b, 90
g, 90
r, 90
ZappyTypes::Colors, 62
Z_BEIGE, 63
Z_BLACK, 63
Z_BLANK, 63
Z_BLUE, 63
Z_BROWN, 63
Z_DARKBLUE, 63
Z_DARKBROWN, 64
Z_DARKGRAY, 64
Z_DARKGREEN, 64
Z_DARKPURPLE, 64
Z_GOLD, 64
Z_GRAY, 64
Z_GREEN, 64
Z_LIGHTGRAY, 64
Z_LIME, 65
Z_MAGENTA, 65
Z_MAROON, 65
Z_ORANGE, 65
Z_PINK, 65
Z_PURPLE, 65
Z_RAYWHITE, 65
Z_RED, 65
Z_SKYBLUE, 65
Z_VIOLET, 66
Z_WHITE, 66
Z_YELLOW, 66
ZappyTypes::Rectangle, 242
height, 242
width, 242
x, 242
y, 243
ZappyTypes::Vector2, 307
x, 308
y, 308
ZappyTypes::Vector3, 308
x, 308
y, 308
z, 308