

Battleship-Online

Generated by Doxygen 1.8.11

Contents

1	LO23-Battleship-Online	1
2	Namespace Index	3
2.1	Packages	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Class Index	7
4.1	Class List	7
5	File Index	11
5.1	File List	11
6	Namespace Documentation	13
6.1	Package data	13
6.2	Package guiMain	13
6.3	Package guiMain.controller	14
6.4	Package guiTable	14
6.5	Package guiTable.controllers	14
6.6	Package interfacesData	15
6.7	Package lo23	15
6.8	Package lo23.battleship	15
6.9	Package lo23.battleship.online	15
6.10	Package lo23.battleship.online.network	16
6.10.1	Detailed Description	16
6.11	Package lo23.battleship.online.network.messages	16
6.11.1	Detailed Description	18
6.12	Package structData	18

7	Class Documentation	19
7.1	guiTable.controllers.BaseController Class Reference	19
7.1.1	Detailed Description	19
7.1.2	Member Function Documentation	19
7.1.2.1	fillChatSlot(AnchorPane pane, String chatFXMLUrl, String conversation)	19
7.1.2.2	fillElement(AnchorPane paneToFill, String contentAdress)	20
7.1.3	Member Data Documentation	20
7.1.3.1	GRID_ELEMENT_SIZE	20
7.1.3.2	GRID_X	20
7.1.3.3	GRID_Y	20
7.1.3.4	NB_CASES_GRID	20
7.1.3.5	SPACE	20
7.2	guiTable.controllers.BelgianPlacementPhaseController Class Reference	21
7.2.1	Detailed Description	21
7.2.2	Member Function Documentation	22
7.2.2.1	initBoatMap()	22
7.2.3	Member Data Documentation	22
7.2.3.1	croiseurRectangle1	22
7.2.3.2	croiseurRectangle2	22
7.2.3.3	cuirasseRectangle	22
7.2.3.4	sousMarinRectangle1	22
7.2.3.5	sousMarinRectangle2	22
7.2.3.6	sousMarinRectangle3	22
7.2.3.7	sousMarinRectangle4	22
7.2.3.8	torpilleurRectangle1	22
7.2.3.9	torpilleurRectangle2	22
7.2.3.10	torpilleurRectangle3	22
7.3	structData.Boat Class Reference	22
7.3.1	Detailed Description	23
7.3.2	Constructor & Destructor Documentation	23

7.3.2.1	Boat()	23
7.3.2.2	Boat(BoatType typedata, Boolean rotation, Position pos)	23
7.3.2.3	Boat(BoatType typedata, Boolean statusdata, List< Position > listCasesdata)	24
7.3.3	Member Function Documentation	24
7.3.3.1	addPosition(Position pos)	24
7.3.3.2	cloneBoat(Boat boatclone)	24
7.3.3.3	getListCases()	24
7.3.3.4	getSunk()	25
7.3.3.5	getType()	25
7.3.3.6	setListcases(List< Position > listCasesdata)	25
7.3.3.7	setListcases(Boolean rotation, Position pos)	25
7.3.3.8	setSunk(Boolean statusdata)	25
7.3.3.9	setType(BoatType typedata)	25
7.3.3.10	updateShot(Position shot)	26
7.3.3.11	verifyBoatStatus()	26
7.3.4	Member Data Documentation	26
7.3.4.1	isSunk	26
7.3.4.2	listCases	26
7.3.4.3	serialVersionUID	26
7.3.4.4	type	26
7.4	guiTable.BoatDrawing Class Reference	26
7.4.1	Detailed Description	27
7.4.2	Constructor & Destructor Documentation	27
7.4.2.1	BoatDrawing(BoatType boatType, Rectangle boatRectangle)	27
7.4.3	Member Function Documentation	28
7.4.3.1	getActiveColor()	28
7.4.3.2	getBadPlacementColor()	28
7.4.3.3	getBoatRectangle()	28
7.4.3.4	getBoatSize()	28
7.4.3.5	getBoatType()	28

7.4.3.6	getDisactiveColor()	28
7.4.3.7	getGridCol()	28
7.4.3.8	getGridRow()	28
7.4.3.9	getInitialLayoutX()	28
7.4.3.10	getInitialLayoutY()	28
7.4.3.11	isActive()	28
7.4.3.12	isPlaced()	28
7.4.3.13	isRotation()	28
7.4.3.14	reinit()	28
7.4.3.15	setActive(boolean active)	28
7.4.3.16	setActiveBoat(HashMap< Rectangle, BoatDrawing > boatMap)	28
7.4.3.17	setBoatRectangle(Rectangle boatRectangle)	29
7.4.3.18	setBoatType(BoatType boatType)	29
7.4.3.19	setPlaced(Boolean bool)	29
7.4.3.20	setPosition(Integer colIndex, Integer rowIndex)	29
7.4.3.21	setRotation(boolean rotation)	29
7.4.4	Member Data Documentation	29
7.4.4.1	active	29
7.4.4.2	activeColor	29
7.4.4.3	badPlacementColor	29
7.4.4.4	boatRectangle	29
7.4.4.5	boatType	29
7.4.4.6	disactiveColor	29
7.4.4.7	gridCol	29
7.4.4.8	gridRow	29
7.4.4.9	initialLayoutX	29
7.4.4.10	initialLayoutY	29
7.4.4.11	placed	29
7.4.4.12	rotation	29
7.5	structData.BoatType Enum Reference	30

7.5.1	Detailed Description	30
7.5.2	Constructor & Destructor Documentation	30
7.5.2.1	BoatType(String name, int nbCases)	30
7.5.3	Member Function Documentation	30
7.5.3.1	getName()	30
7.5.3.2	getNbCases()	31
7.5.4	Member Data Documentation	31
7.5.4.1	CONTRETORPILLEUR	31
7.5.4.2	CROISEURB	31
7.5.4.3	CROISEURFR	31
7.5.4.4	CUIRASSE	31
7.5.4.5	name	31
7.5.4.6	nbCases	31
7.5.4.7	PORTEAVIONS	31
7.5.4.8	SOUSMARINB	31
7.5.4.9	SOUSMARINFR	31
7.5.4.10	TORPILLEUR	31
7.6	guiTable.CaseDrawing Class Reference	31
7.6.1	Detailed Description	32
7.6.2	Constructor & Destructor Documentation	32
7.6.2.1	CaseDrawing()	32
7.6.2.2	CaseDrawing(Type t)	32
7.7	data.CDataCom Class Reference	32
7.7.1	Detailed Description	34
7.7.2	Constructor & Destructor Documentation	34
7.7.2.1	CDataCom(DataController dc)	34
7.7.3	Member Function Documentation	34
7.7.3.1	addNewGameList(Game g)	34
7.7.3.2	addUserToUserList(User u)	34
7.7.3.3	changeStatusGame(Game g)	35

7.7.3.4	<code>coordinates(Shot s)</code>	35
7.7.3.5	<code>coordinates(Shot s, Boat b)</code>	35
7.7.3.6	<code>getCreatedGame()</code>	35
7.7.3.7	<code>getLocalUser()</code>	36
7.7.3.8	<code>getOtherPlayer()</code>	36
7.7.3.9	<code>getPlayerPosition(Player p, Game g)</code>	36
7.7.3.10	<code>getUserProfile()</code>	36
7.7.3.11	<code>joinGameSpectator(Game g)</code>	36
7.7.3.12	<code>newRequestSpectator(User u)</code>	37
7.7.3.13	<code>notifiedGameWon()</code>	37
7.7.3.14	<code>notifToJoinGame(Profile sender, Game g)</code>	37
7.7.3.15	<code>notifyAttendedGameWon(Player p)</code>	37
7.7.3.16	<code>notifyQuitSpectator(User spec)</code>	38
7.7.3.17	<code>notifyToSpecGame(User spec)</code>	38
7.7.3.18	<code>receiveMessage(ChatMessage message)</code>	38
7.7.3.19	<code>receiveReady()</code>	38
7.7.3.20	<code>removeGame(Game g)</code>	38
7.7.3.21	<code>removeGameFromList(Game g)</code>	39
7.7.3.22	<code>removeUser(User u)</code>	39
7.7.3.23	<code>sendStatistics(Profile profile)</code>	39
7.7.3.24	<code>setGameJoinResponse(Boolean ok, Player player1, Player player2)</code>	39
7.7.3.25	<code>setGameJoinResponse(Boolean no)</code>	39
7.7.3.26	<code>setInterfaceCom(COMInterface c)</code>	40
7.7.3.27	<code>setInterfaceMain(GuiMainInterface i)</code>	40
7.7.3.28	<code>setInterfaceTable(GuiTableInterface i)</code>	40
7.7.3.29	<code>setLocalGame(Game g)</code>	40
7.7.3.30	<code>updateAttendedGame(Player p, Shot s, Boat b)</code>	40
7.7.4	Member Data Documentation	40
7.7.4.1	<code>controller</code>	40
7.7.4.2	<code>interfaceCom</code>	40

7.7.4.3	interfaceMain	40
7.7.4.4	interfaceTable	40
7.8	data.CDataMain Class Reference	41
7.8.1	Detailed Description	42
7.8.2	Constructor & Destructor Documentation	42
7.8.2.1	CDataMain(DataController dc)	42
7.8.3	Member Function Documentation	42
7.8.3.1	askDisconnection()	42
7.8.3.2	clear()	42
7.8.3.3	connection(String login, String password)	42
7.8.3.4	createAccount(String login, String username, HashSet ips, int port, String password, List< ContactGroup > contactList, String avatar, String lastname, String firstname, Date birthDate)	43
7.8.3.5	editProfile(String username, String password, String avatar, String lastName, String firstName, Date birthDate)	43
7.8.3.6	gameToSpec(Game g)	43
7.8.3.7	getGames()	43
7.8.3.8	getListUsers()	44
7.8.3.9	getLocalProfile()	44
7.8.3.10	getProfile(User u)	44
7.8.3.11	newGame(Boolean newClassicType, String newName, Boolean newHuman↔ Opponent, Integer newTimePerShot, Integer newTimeToPlaceBoats, Boolean newSpectator, Boolean newSpectatorChat)	44
7.8.3.12	notifGameChosen(Game g)	45
7.8.3.13	removeGame(Game g)	45
7.8.3.14	setInterfaceCom(COMInterface i)	45
7.8.3.15	setListIps(HashSet Ips)	45
7.8.3.16	setLocalGame(Game g)	45
7.8.3.17	setPort(int p)	46
7.8.4	Member Data Documentation	46
7.8.4.1	controller	46
7.8.4.2	interfaceCom	46
7.9	data.CDataTable Class Reference	46

7.9.1	Detailed Description	47
7.9.2	Constructor & Destructor Documentation	47
7.9.2.1	CDataTable(DataController dc)	47
7.9.3	Member Function Documentation	47
7.9.3.1	changeStatusGameStarted()	47
7.9.3.2	coordinate(Position pos)	47
7.9.3.3	coordinateShips(List< Boat > listBoat)	48
7.9.3.4	exit()	48
7.9.3.5	gameEnded()	48
7.9.3.6	getLocalGame()	48
7.9.3.7	getObserverGame()	48
7.9.3.8	setInterfaceCom(COMInterface c)	48
7.9.3.9	setInterfaceMain(GuiMainInterface m)	49
7.9.3.10	setInterfaceTable(GuiTableInterface t)	49
7.9.3.11	textMessage(String message)	49
7.9.3.12	timerOver()	49
7.9.4	Member Data Documentation	49
7.9.4.1	controller	49
7.9.4.2	interfaceCom	49
7.9.4.3	interfaceMain	49
7.9.4.4	interfaceTable	50
7.9.4.5	NB_CASES_GRID	50
7.10	guiMain.controller.ChangeProfileController Class Reference	50
7.10.1	Member Function Documentation	51
7.10.1.1	backToMenu(ActionEvent event)	51
7.10.1.2	init(User user)	51
7.10.1.3	initialize(URL location, ResourceBundle resources)	51
7.10.1.4	modifyAvatar(ActionEvent event)	51
7.10.1.5	openFile(File file)	52
7.10.1.6	saveProfile(ActionEvent event)	52

7.10.1.7	setMainController(GuiMainController c)	52
7.10.2	Member Data Documentation	52
7.10.2.1	avatarPath	52
7.10.2.2	backButton	52
7.10.2.3	birthdate	52
7.10.2.4	errorMessage	52
7.10.2.5	firstName	52
7.10.2.6	lastName	52
7.10.2.7	mainController	52
7.10.2.8	modifyAvatarButton	52
7.10.2.9	nameTitle	52
7.10.2.10	numberOfGame	53
7.10.2.11	numberOfGameLost	53
7.10.2.12	numberOfGameWon	53
7.10.2.13	saveButton	53
7.10.2.14	userAvatar	53
7.10.2.15	userName	53
7.10.2.16	userPassword	53
7.11	guiTable.controllers.ChatController Class Reference	53
7.11.1	Member Function Documentation	53
7.11.1.1	createConversation(ArrayList< ChatMessage > list)	53
7.11.1.2	doProfileArea()	54
7.11.1.3	getConversation()	54
7.11.1.4	init(String conv)	54
7.11.1.5	receiveAMessage(ChatMessage message)	54
7.11.1.6	reloadConversation()	54
7.11.1.7	sendMyMessage()	54
7.11.1.8	setDataController(IDataTable d)	54
7.11.2	Member Data Documentation	54
7.11.2.1	conversation	54

7.11.2.2	conversationArea	54
7.11.2.3	dataController	54
7.11.2.4	field	54
7.11.2.5	profil	54
7.12	structData.ChatMessage Class Reference	54
7.12.1	Detailed Description	55
7.12.2	Constructor & Destructor Documentation	55
7.12.2.1	ChatMessage(User newSender, String newMessage, Date newBirthdate)	55
7.12.3	Member Function Documentation	56
7.12.3.1	cloneChatMessage(ChatMessage chatMessageClone)	56
7.12.3.2	getContent()	56
7.12.3.3	getProfile()	56
7.12.3.4	getTime()	56
7.12.3.5	setContent(String m)	56
7.12.3.6	setProfile(User u)	57
7.12.3.7	setTime(Date t)	57
7.12.4	Member Data Documentation	57
7.12.4.1	content	57
7.12.4.2	profile	57
7.12.4.3	serialVersionUID	57
7.12.4.4	time	57
7.13	guiTable.controllers.ClassicPlacementPhaseController Class Reference	57
7.13.1	Detailed Description	58
7.13.2	Member Function Documentation	58
7.13.2.1	initBoatMap()	58
7.13.3	Member Data Documentation	58
7.13.3.1	contreTorpilleurRectangle	58
7.13.3.2	croiseurRectangle	58
7.13.3.3	porteAvionsRectangle	58
7.13.3.4	sousMarinRectangle	58

7.13.3.5	torpilleurRectangle	58
7.14	Io23.battleship.online.network.COMInterface Interface Reference	58
7.14.1	Detailed Description	59
7.14.2	Member Function Documentation	59
7.14.2.1	askDisconnection()	59
7.14.2.2	changeStatusGame(Game game)	60
7.14.2.3	clearNetwork()	60
7.14.2.4	coordinates(Player destPlayer, Shot resultShot, Game game, Boat boat)	60
7.14.2.5	gameQuitSpectator(User spec, Game game)	60
7.14.2.6	getInfoGameForSpectator(Player player, User spec)	60
7.14.2.7	getProfile(User user)	61
7.14.2.8	joinGame(Game g)	61
7.14.2.9	notifyGameWon()	61
7.14.2.10	notifyJoinGameResponse(boolean isOk, Profile user, Game g)	61
7.14.2.11	notifyNewGame(Game g)	62
7.14.2.12	notifyReady(User user, Player playerToNotify)	62
7.14.2.13	removeGame(Game game)	62
7.14.2.14	searchForPlayers()	62
7.14.2.15	sendChatMessage(ChatMessage chatMessage, Game g)	62
7.14.2.16	sendInfoGameForSpectator(Game game, User spec)	63
7.14.2.17	sendNewSpectator(User u, Player p, HashSet< User > listSpectator)	63
7.14.2.18	sendShot(Player player, Game g, Shot shot)	63
7.15	Io23.battleship.online.network.messages.ConnectionEstablishedMessage Class Reference	64
7.15.1	Detailed Description	64
7.15.2	Constructor & Destructor Documentation	65
7.15.2.1	ConnectionEstablishedMessage(User sender, List< InetAddress > ipTable, Game game)	65
7.15.3	Member Function Documentation	65
7.15.3.1	getType()	65
7.15.3.2	process(IDataCom IData, InetAddress senderAddress)	65
7.15.4	Member Data Documentation	66

7.15.4.1	createdGame	66
7.15.4.2	ipAddressesTable	66
7.15.4.3	sender	66
7.16	Io23.battleship.online.network.messages.ConnectionRequestMessage Class Reference	66
7.16.1	Detailed Description	67
7.16.2	Constructor & Destructor Documentation	67
7.16.2.1	ConnectionRequestMessage(User user, List< InetAddress > ipAddressesTable, Game game)	67
7.16.3	Member Function Documentation	67
7.16.3.1	getType()	67
7.16.3.2	process(IDataCom IData, InetAddress senderAddress)	67
7.16.4	Member Data Documentation	68
7.16.4.1	game	68
7.16.4.2	ipAddressesTable	68
7.16.4.3	sender	68
7.17	structData.ContactGroup Class Reference	68
7.17.1	Detailed Description	68
7.17.2	Constructor & Destructor Documentation	69
7.17.2.1	ContactGroup()	69
7.17.3	Member Data Documentation	69
7.17.3.1	listUsers	69
7.17.3.2	name	69
7.17.3.3	rights	69
7.17.3.4	serialVersionUID	69
7.18	Io23.battleship.online.network.messages.CreatedGameNotificationMessage Class Reference	69
7.18.1	Detailed Description	70
7.18.2	Constructor & Destructor Documentation	70
7.18.2.1	CreatedGameNotificationMessage(Game gameCreated)	70
7.18.3	Member Function Documentation	70
7.18.3.1	getType()	70
7.18.3.2	process(IDataCom IData, InetAddress senderAddress)	70

7.18.4	Member Data Documentation	71
7.18.4.1	game	71
7.19	guiMain.controller.CreateGameController Class Reference	71
7.19.1	Member Function Documentation	72
7.19.1.1	initialize(URL url, ResourceBundle rb)	72
7.19.1.2	newGame(ActionEvent event)	73
7.19.1.3	returnToMenu(ActionEvent event)	73
7.19.1.4	setMainController(GuiMainController c)	73
7.19.1.5	timerDisable(ActionEvent event)	73
7.19.2	Member Data Documentation	73
7.19.2.1	belgiumGameType	74
7.19.2.2	chatAutorise	74
7.19.2.3	classicGameType	74
7.19.2.4	createGame	74
7.19.2.5	errorMessage	74
7.19.2.6	gameName	74
7.19.2.7	gameType	74
7.19.2.8	mainController	74
7.19.2.9	positioningTime	74
7.19.2.10	reflectionTime	74
7.19.2.11	returnButton	74
7.19.2.12	spectatorsAutorise	74
7.19.2.13	timeDisabled	74
7.20	data.DataController Class Reference	74
7.20.1	Detailed Description	77
7.20.2	Constructor & Destructor Documentation	77
7.20.2.1	DataController()	77
7.20.3	Member Function Documentation	77
7.20.3.1	addGameToList(Game g)	77
7.20.3.2	addUserToList(User u)	77

7.20.3.3	<code>clearData()</code>	77
7.20.3.4	<code>endGame()</code>	77
7.20.3.5	<code>gameOver()</code>	78
7.20.3.6	<code>getAttendedGame()</code>	78
7.20.3.7	<code>getInterfaceDataCom()</code>	78
7.20.3.8	<code>getInterfaceDataMain()</code>	78
7.20.3.9	<code>getInterfaceDataTable()</code>	78
7.20.3.10	<code>getListGames()</code>	78
7.20.3.11	<code>getListUsers()</code>	79
7.20.3.12	<code>getLocalDataUser()</code>	79
7.20.3.13	<code>getLocalGame()</code>	79
7.20.3.14	<code>getLocalPlayer()</code>	79
7.20.3.15	<code>getLocalPlayerInGame()</code>	79
7.20.3.16	<code>getLocalProfile()</code>	79
7.20.3.17	<code>getLocalUser()</code>	80
7.20.3.18	<code>getOtherPPlayer()</code>	80
7.20.3.19	<code>getPlayerPosition(Player p, Game g)</code>	80
7.20.3.20	<code>getTableInterface()</code>	80
7.20.3.21	<code>immediateDefeat()</code>	80
7.20.3.22	<code>isPlayer1()</code>	80
7.20.3.23	<code>isPlayerOf(Game game)</code>	81
7.20.3.24	<code>recordVictory()</code>	81
7.20.3.25	<code>reloadSavedProfile(String login, String mdp)</code>	81
7.20.3.26	<code>removeGameFromList(Game g)</code>	81
7.20.3.27	<code>removeUserFromList(User u)</code>	81
7.20.3.28	<code>setAttendedGame(Game g)</code>	82
7.20.3.29	<code>setInterfaceCom(COMInterface i)</code>	82
7.20.3.30	<code>setInterfaceMain(GuiMainInterface i)</code>	82
7.20.3.31	<code>setInterfaceTable(GuiTableInterface i)</code>	82
7.20.3.32	<code>setListGame(List< Game > g)</code>	82

7.20.3.33	setListUser(List< User > u)	82
7.20.3.34	setLocalDataUser(DataUser du)	83
7.20.3.35	setLocalGame(Game g)	83
7.20.3.36	setLocalPlayer(Player p)	83
7.20.3.37	setLocalProfile(Profile p)	83
7.20.3.38	setLocalUser(User u)	83
7.20.3.39	testShot(Shot s)	84
7.20.3.40	updateGameData(Boolean ok, Player player1, Player player2)	84
7.20.3.41	updateGameDataPlaying(Shot s, Boat b, boolean forLocalPlayer)	84
7.20.3.42	updateGameStatus(Game g)	84
7.20.4	Member Data Documentation	84
7.20.4.1	attendedGame	84
7.20.4.2	interfaceCom	85
7.20.4.3	interfaceDataCom	85
7.20.4.4	interfaceDataMain	85
7.20.4.5	interfaceDataTable	85
7.20.4.6	interfaceMain	85
7.20.4.7	interfaceTable	85
7.20.4.8	listGames	85
7.20.4.9	listUsers	85
7.20.4.10	localGame	85
7.20.4.11	localPlayer	85
7.20.4.12	localProfile	85
7.20.4.13	localUser	85
7.21	structData.DataUser Class Reference	85
7.21.1	Detailed Description	86
7.21.2	Constructor & Destructor Documentation	86
7.21.2.1	DataUser(User u)	86
7.21.2.2	DataUser(User u, String passworddata, List< ContactGroup > listContactsdata)	86
7.21.2.3	DataUser(DataUser du)	87

7.21.3	Member Function Documentation	87
7.21.3.1	cloneDataUser(DataUser duserclone)	87
7.21.3.2	getListContacts()	87
7.21.3.3	getPassword()	87
7.21.3.4	setListContacts(List< ContactGroup > listContactsdata)	87
7.21.3.5	setPassword(String passworddata)	88
7.21.4	Member Data Documentation	88
7.21.4.1	listContacts	88
7.21.4.2	password	88
7.22	Io23.battleship.online.network.messages.DisconnectionMessage Class Reference	88
7.22.1	Detailed Description	89
7.22.2	Constructor & Destructor Documentation	89
7.22.2.1	DisconnectionMessage(User userWhoDisconnect, Game game)	89
7.22.3	Member Function Documentation	89
7.22.3.1	getType()	89
7.22.3.2	process(IDataCom IData, InetAddress senderAddress)	90
7.22.4	Member Data Documentation	90
7.22.4.1	game	90
7.22.4.2	user	90
7.23	structData.Game Class Reference	90
7.23.1	Detailed Description	92
7.23.2	Constructor & Destructor Documentation	92
7.23.2.1	Game(Profile p)	92
7.23.2.2	Game(Boolean newClassicType, String newName, Boolean newHuman↔ Opponent, Integer newTimePerShot, Integer newTimeToPlaceBoats, Boolean newSpectator, Boolean newSpectatorChat, Profile p)	92
7.23.2.3	Game(Boolean newClassicType, String newName, Boolean newHuman↔ Opponent, Integer newTimePerShot, Integer newTimeToPlaceBoats, Boolean newSpectator, Boolean newSpectatorChat, Profile p, HashSet IS, Player p1, Player p2, Boolean p1Start, ArrayList lMsg)	93
7.23.2.4	Game(Game g)	93
7.23.3	Member Function Documentation	93
7.23.3.1	addMessage(ChatMessage msg)	93

7.23.3.2	addSpectators(User spectator)	94
7.23.3.3	clone(Game g)	94
7.23.3.4	doesProfileBelongToGame(Profile p)	94
7.23.3.5	getClassicType()	94
7.23.3.6	getHumanOpponent()	94
7.23.3.7	getIdGame()	95
7.23.3.8	getListMessages()	95
7.23.3.9	getListSpectators()	95
7.23.3.10	getName()	95
7.23.3.11	getPlayer1()	95
7.23.3.12	getPlayer1Start()	95
7.23.3.13	getPlayer2()	96
7.23.3.14	getSpectator()	96
7.23.3.15	getSpectatorChat()	96
7.23.3.16	getStatus()	96
7.23.3.17	getTimePerShot()	96
7.23.3.18	getTimeToPlaceBoats()	96
7.23.3.19	setListMessages(ArrayList lMsg)	96
7.23.3.20	setListSpectators(HashSet< User > IS)	97
7.23.3.21	setPlayer1(Player p1)	97
7.23.3.22	setPlayer2(Player p2)	97
7.23.3.23	setStatus(StatusGame status)	97
7.23.4	Member Data Documentation	97
7.23.4.1	classicType	97
7.23.4.2	humanOpponent	97
7.23.4.3	idGame	98
7.23.4.4	listMessages	98
7.23.4.5	listSpectators	98
7.23.4.6	name	98
7.23.4.7	player1	98

7.23.4.8	player1Start	98
7.23.4.9	player2	98
7.23.4.10	serialVersionUID	98
7.23.4.11	spectator	98
7.23.4.12	spectatorChat	98
7.23.4.13	status	98
7.23.4.14	timePerShot	98
7.23.4.15	timeToPlaceBoats	98
7.24	guiMain.GameCell Class Reference	98
7.24.1	Detailed Description	99
7.24.2	Constructor & Destructor Documentation	99
7.24.2.1	GameCell(menuController c)	99
7.24.3	Member Function Documentation	99
7.24.3.1	updateItem(final Game game, boolean empty)	99
7.24.4	Member Data Documentation	100
7.24.4.1	controller	100
7.24.4.2	hbox	100
7.24.4.3	joinButton	100
7.24.4.4	joinButtonBox	100
7.24.4.5	lookButton	100
7.24.4.6	lookButtonBox	100
7.24.4.7	robot	100
7.24.4.8	title	100
7.24.4.9	type	100
7.25	guiTable.controllers.gameInterface Class Reference	100
7.25.1	Detailed Description	101
7.25.2	Member Function Documentation	101
7.25.2.1	logMsg(String msg)	101
7.25.2.2	logYesNoMsg(String msg)	102
7.25.2.3	placeShotTo(Shot shot, GridPane grid)	102

7.25.2.4	showDefeat()	102
7.25.2.5	showVictory()	102
7.25.2.6	sunkABoat(GridPane grid, Boat boat)	102
7.25.2.7	yesClicked(ActionEvent event)	102
7.25.3	Member Data Documentation	102
7.25.3.1	anchorPane	102
7.25.3.2	chatPane	102
7.25.3.3	DEFEAT_MSG	102
7.25.3.4	EXIT_GAME_MSG	102
7.25.3.5	exitButton	102
7.25.3.6	messageContainer	102
7.25.3.7	messageTextContainer	103
7.25.3.8	MY_TURN_MSG	103
7.25.3.9	noButton	103
7.25.3.10	OTHER_TURN_MSG	103
7.25.3.11	STYLE_MY_TURN	103
7.25.3.12	STYLE_OTHER_TURN	103
7.25.3.13	tableController	103
7.25.3.14	VICTORY_MSG	103
7.25.3.15	waitExit	103
7.25.3.16	waitMsg	103
7.25.3.17	yesButton	103
7.26	guiTable.controllers.GamePhaseController Class Reference	103
7.26.1	Detailed Description	105
7.26.2	Member Function Documentation	105
7.26.2.1	addOpponentShot(Shot opponentShot)	105
7.26.2.2	addShot(Shot shot)	105
7.26.2.3	exitGame()	105
7.26.2.4	getChatPane()	105
7.26.2.5	initialize(URL location, ResourceBundle resources)	105

7.26.2.6	<code>noClicked()</code>	105
7.26.2.7	<code>onClickCase()</code>	106
7.26.2.8	<code>removeSelectedCase()</code>	106
7.26.2.9	<code>setMyBoats(List< Boat > boats)</code>	106
7.26.2.10	<code>setMyTurn(Boolean myTurn)</code>	106
7.26.2.11	<code>setRoundTime()</code>	106
7.26.2.12	<code>showDefeat()</code>	106
7.26.2.13	<code>showVictory()</code>	106
7.26.2.14	<code>sunkBoat(Boat boat)</code>	106
7.26.2.15	<code>sunkMyBoat(Boat boat)</code>	107
7.26.2.16	<code>timelsOver()</code>	107
7.26.2.17	<code>validateShot()</code>	107
7.26.3	Member Data Documentation	107
7.26.3.1	<code>gameState</code>	107
7.26.3.2	<code>messageMask</code>	107
7.26.3.3	<code>myTable</code>	107
7.26.3.4	<code>myTurn</code>	107
7.26.3.5	<code>selectedCase</code>	107
7.26.3.6	<code>table</code>	107
7.26.3.7	<code>time</code>	107
7.26.3.8	<code>timeline</code>	107
7.26.3.9	<code>timePerShot</code>	107
7.26.3.10	<code>timerLabel</code>	107
7.26.3.11	<code>valider</code>	107
7.27	<code>lo23.battleship.online.network.messages.GameQuitMessage</code> Class Reference	108
7.27.1	Detailed Description	108
7.27.2	Constructor & Destructor Documentation	109
7.27.2.1	<code>GameQuitMessage(Game game)</code>	109
7.27.3	Member Function Documentation	109
7.27.3.1	<code>getType()</code>	109

7.27.3.2	process(IDataCom IData, InetAddress senderAddress)	109
7.27.4	Member Data Documentation	109
7.27.4.1	game	109
7.28	Io23.battleship.online.network.messages.GameQuitSpectatorMessage Class Reference	109
7.28.1	Detailed Description	110
7.28.2	Constructor & Destructor Documentation	110
7.28.2.1	GameQuitSpectatorMessage(Game g, User s)	110
7.28.3	Member Function Documentation	110
7.28.3.1	getType()	110
7.28.3.2	process(IDataCom IData, InetAddress senderAddress)	111
7.28.4	Member Data Documentation	111
7.28.4.1	game	111
7.28.4.2	spec	111
7.29	Io23.battleship.online.network.messages.GameWonMessage Class Reference	111
7.29.1	Detailed Description	112
7.29.2	Constructor & Destructor Documentation	112
7.29.2.1	GameWonMessage(Player p)	112
7.29.3	Member Function Documentation	112
7.29.3.1	getType()	112
7.29.3.2	process(IDataCom IData, InetAddress senderAddress)	112
7.29.4	Member Data Documentation	113
7.29.4.1	player	113
7.30	Io23.battleship.online.network.messages.GameWonMessageToSpectator Class Reference	113
7.30.1	Detailed Description	114
7.30.2	Constructor & Destructor Documentation	114
7.30.2.1	GameWonMessageToSpectator(Player p)	114
7.30.3	Member Function Documentation	114
7.30.3.1	getType()	114
7.30.3.2	process(IDataCom IData, InetAddress senderAddress)	114
7.30.4	Member Data Documentation	115

7.30.4.1	player	115
7.31	lo23.battleship.online.network.messages.GetInfoGameForSpectatorMessage Class Reference	115
7.31.1	Detailed Description	115
7.31.2	Constructor & Destructor Documentation	116
7.31.2.1	GetInfoGameForSpectatorMessage(Player p, User s)	116
7.31.3	Member Function Documentation	116
7.31.3.1	getType()	116
7.31.3.2	process(IDataCom IData, InetAddress senderAddress)	116
7.31.4	Member Data Documentation	116
7.31.4.1	player	116
7.31.4.2	spec	116
7.32	lo23.battleship.online.network.messages.GetProfileRequestMessage Class Reference	117
7.32.1	Detailed Description	117
7.32.2	Constructor & Destructor Documentation	117
7.32.2.1	GetProfileRequestMessage(User requester)	117
7.32.3	Member Function Documentation	118
7.32.3.1	getType()	118
7.32.3.2	process(IDataCom IData, InetAddress senderAddress)	118
7.32.4	Member Data Documentation	118
7.32.4.1	userRequester	118
7.33	guiMain.GuiMainController Class Reference	119
7.33.1	Detailed Description	120
7.33.2	Constructor & Destructor Documentation	120
7.33.2.1	GuiMainController(Stage s)	120
7.33.3	Member Function Documentation	121
7.33.3.1	addGame(final Game createdGame)	121
7.33.3.2	addUser(final User user)	121
7.33.3.3	askJoinGame(final Game game)	121
7.33.3.4	getIdata()	121
7.33.3.5	getIps()	121

7.33.3.6	getPort()	122
7.33.3.7	lookGame(Game game)	122
7.33.3.8	openChangeProfileWindow(User user)	122
7.33.3.9	openConfigWindow()	122
7.33.3.10	openCreateGameWindow()	122
7.33.3.11	openMenuWindow()	122
7.33.3.12	openPlacementPhase(final Game game)	122
7.33.3.13	openProfileWindow(User user)	122
7.33.3.14	openSignupWindow()	123
7.33.3.15	openWaitingRoomWindow(Game game)	123
7.33.3.16	removeGame(final Game removedGame)	123
7.33.3.17	removeUser(final User user)	123
7.33.3.18	sendStatistics(Profile profil)	123
7.33.3.19	setGameJoinResponse(boolean isOk)	124
7.33.3.20	setIdata(IDataMain idata)	124
7.33.3.21	setIps(List< String > list)	124
7.33.3.22	setPort(int num_port)	124
7.33.3.23	startIHM()	124
7.33.3.24	transmitNewStatus(final Game game)	124
7.33.4	Member Data Documentation	125
7.33.4.1	changeProfileController	125
7.33.4.2	createGameController	125
7.33.4.3	gamesList	125
7.33.4.4	idata	125
7.33.4.5	ipConfigController	125
7.33.4.6	ipsList	125
7.33.4.7	loginController	125
7.33.4.8	menuController	125
7.33.4.9	playersList	125
7.33.4.10	port	125

7.33.4.11	profilController	125
7.33.4.12	rootLayout	125
7.33.4.13	signUpController	125
7.33.4.14	stage	125
7.33.4.15	waitingRoomController	125
7.34	guiMain.GuiMainInterface Interface Reference	126
7.34.1	Detailed Description	126
7.34.2	Member Function Documentation	126
7.34.2.1	addGame(Game createdGame)	126
7.34.2.2	addUser(User user)	127
7.34.2.3	openMenuWindow()	127
7.34.2.4	openPlacementPhase(Game game)	127
7.34.2.5	removeGame(Game game)	127
7.34.2.6	removeUser(User user)	127
7.34.2.7	sendStatistics(Profile profil)	127
7.34.2.8	setGameJoinResponse(boolean isOk)	128
7.34.2.9	transmitNewStatus(Game game)	128
7.35	guiTable.controllers.GuiTableController Class Reference	128
7.35.1	Detailed Description	130
7.35.2	Constructor & Destructor Documentation	130
7.35.2.1	GuiTableController()	130
7.35.3	Member Function Documentation	130
7.35.3.1	addChatMessage(ChatMessage message)	130
7.35.3.2	displayDefeat()	130
7.35.3.3	displayMyShotResult(final Shot myShotResult, final Boat boat)	130
7.35.3.4	displayObserverPhase(Stage currentStage, Game game)	130
7.35.3.5	displayObserverPhaseVictory(int winner)	130
7.35.3.6	displayObserverShot(final Shot shot, int player)	131
7.35.3.7	displayOpponentShot(final Shot opponentShot, final Boat boat)	131
7.35.3.8	displayPlacementPhase(Stage currentStage, Boolean classic, int placementTime)	131

7.35.3.9	displayRageQuit()	131
7.35.3.10	displayVictory()	131
7.35.3.11	exitGame()	132
7.35.3.12	getDataController()	132
7.35.3.13	getInstance()	132
7.35.3.14	opponentReady(final Boolean myTurn, long timePerShot)	132
7.35.3.15	setDataController(CDataTable data)	132
7.35.3.16	sunkPlayerBoat(int i, Boat boat)	132
7.35.3.17	updateSpectatorGame(Game game)	132
7.35.3.18	validateBoats(List< Boat > boats)	132
7.35.3.19	validateShot(Position pos)	132
7.35.4	Member Data Documentation	132
7.35.4.1	BELGE_PLACEMENT_URL	132
7.35.4.2	boats	132
7.35.4.3	CHAT_FXML_URL	132
7.35.4.4	chatController	132
7.35.4.5	classic	132
7.35.4.6	CLASSIC_PLACEMENT_URL	133
7.35.4.7	dataController	133
7.35.4.8	GAME_PHASE_URL	133
7.35.4.9	gamePhaseController	133
7.35.4.10	INSTANCE	133
7.35.4.11	mainStage	133
7.35.4.12	OBSERVATION_PHASE_URL	133
7.35.4.13	observationControlleur	133
7.35.4.14	OBSERVER_PHASE_URL	133
7.35.4.15	observerPhaseController	133
7.35.4.16	placementPhaseController	133
7.35.4.17	rootLayout	133
7.35.4.18	TITLE	133

7.36	guiTable.GuiTableInterface Interface Reference	133
7.36.1	Detailed Description	134
7.36.2	Member Function Documentation	134
7.36.2.1	addChatMessage(ChatMessage message)	134
7.36.2.2	displayDefeat()	134
7.36.2.3	displayMyShotResult(Shot myShotResult, Boat boat)	135
7.36.2.4	displayObserverPhase(Stage currentStage, Game game)	135
7.36.2.5	displayObserverPhaseVictory(int winner)	135
7.36.2.6	displayObserverShot(Shot shot, int player)	135
7.36.2.7	displayOpponentShot(Shot opponentShot, Boat boat)	136
7.36.2.8	displayPlacementPhase(Stage currentStage, Boolean classic, int timePerShot)	136
7.36.2.9	displayRageQuit()	136
7.36.2.10	displayVictory()	136
7.36.2.11	opponentReady(Boolean myTurn, long time)	136
7.36.2.12	setDataController(CDataTable d)	137
7.36.2.13	sunkPlayerBoat(int i, Boat boat)	137
7.36.2.14	updateSpectatorGame(Game g)	137
7.37	interfacesData.IDataCom Interface Reference	137
7.37.1	Detailed Description	138
7.37.2	Member Function Documentation	138
7.37.2.1	addNewGameList(Game g)	138
7.37.2.2	addUserToUserList(User u)	139
7.37.2.3	changeStatusGame(Game g)	139
7.37.2.4	coordinates(Shot s, Boat b)	139
7.37.2.5	coordinates(Shot s)	139
7.37.2.6	getCreatedGame()	140
7.37.2.7	getLocalUser()	140
7.37.2.8	getOtherPlayer()	140
7.37.2.9	getPlayerPosition(Player p, Game g)	140
7.37.2.10	getUserProfile()	141

7.37.2.11 joinGameSpectator(Game g)	141
7.37.2.12 newRequestSpectator(User u)	141
7.37.2.13 notifiedGameWon()	141
7.37.2.14 notifToJoinGame(Profile sender, Game g)	141
7.37.2.15 notifyAttendedGameWon(Player p)	142
7.37.2.16 notifyQuitSpectator(User spec)	142
7.37.2.17 notifyToSpecGame(User spec)	142
7.37.2.18 receiveMessage(ChatMessage message)	142
7.37.2.19 receiveReady()	142
7.37.2.20 removeGame(Game g)	143
7.37.2.21 removeGameFromList(Game g)	143
7.37.2.22 removeUser(User u)	143
7.37.2.23 sendStatistics(Profile p)	143
7.37.2.24 setGameJoinResponse(Boolean ok, Player player1, Player player2)	144
7.37.2.25 setGameJoinResponse(Boolean no)	144
7.37.2.26 setLocalGame(Game g)	144
7.37.2.27 updateAttendedGame(Player p, Shot s, Boat b)	144
7.38 interfacesData.IDataMain Interface Reference	145
7.38.1 Detailed Description	146
7.38.2 Member Function Documentation	146
7.38.2.1 askDisconnection()	146
7.38.2.2 clear()	146
7.38.2.3 connection(String login, String password)	146
7.38.2.4 createAccount(String login, String username, HashSet ips, int port, String password, List< ContactGroup > contactList, String avatar, String lastname, String firstname, Date birthDate)	146
7.38.2.5 editProfile(String username, String password, String avatar, String lastName, String firstName, Date borthDate)	147
7.38.2.6 gameToSpec(Game g)	147
7.38.2.7 getGames()	147
7.38.2.8 getListUsers()	148
7.38.2.9 getLocalProfile()	148

7.38.2.10	getProfile(User u)	148
7.38.2.11	newGame(Boolean newClassicType, String newName, Boolean newHuman↔ Opponent, Integer newTimePerShot, Integer newTimeToPlaceBoats, Boolean newSpectator, Boolean newSpectatorChat)	148
7.38.2.12	notifGameChosen(Game g)	149
7.38.2.13	removeGame(Game g)	149
7.38.2.14	setListIps(HashSet Ips)	149
7.38.2.15	setLocalGame(Game g)	149
7.38.2.16	setPort(int p)	149
7.39	interfacesData.IDataTable Interface Reference	150
7.39.1	Detailed Description	150
7.39.2	Member Function Documentation	151
7.39.2.1	changeStatusGameStarted()	151
7.39.2.2	coordinate(Position pos)	151
7.39.2.3	coordinateShips(List< Boat > listBoat)	151
7.39.2.4	exit()	151
7.39.2.5	gameEnded()	151
7.39.2.6	getLocalGame()	152
7.39.2.7	getObserverGame()	152
7.39.2.8	textMessage(String message)	152
7.39.2.9	timerOver()	152
7.40	guiMain.IpCell Class Reference	152
7.40.1	Detailed Description	153
7.40.2	Constructor & Destructor Documentation	153
7.40.2.1	IpCell()	153
7.40.3	Member Function Documentation	153
7.40.3.1	updateItem(String item, boolean empty)	153
7.40.4	Member Data Documentation	153
7.40.4.1	hbox	154
7.40.4.2	ipLabel	154
7.40.4.3	pane	154

7.40.4.4	removeButton	154
7.41	guiMain.controller.IpConfigController Class Reference	154
7.41.1	Detailed Description	154
7.41.2	Member Function Documentation	155
7.41.2.1	addIp()	155
7.41.2.2	init()	155
7.41.2.3	initIpsList()	155
7.41.2.4	setMainController(GuiMainController c)	155
7.41.2.5	validate(ActionEvent event)	155
7.41.3	Member Data Documentation	155
7.41.3.1	addButton	155
7.41.3.2	ipsListView	155
7.41.3.3	ipTextField	155
7.41.3.4	mainController	155
7.41.3.5	port	155
7.41.3.6	validateButton	155
7.42	Io23.battleship.online.network.messages.JoinGameRequestMessage Class Reference	156
7.42.1	Detailed Description	156
7.42.2	Constructor & Destructor Documentation	156
7.42.2.1	JoinGameRequestMessage(Profile senderGame, Game gameSend)	156
7.42.3	Member Function Documentation	157
7.42.3.1	getType()	157
7.42.3.2	process(IDataCom IData, InetAddress senderAddress)	157
7.42.4	Member Data Documentation	157
7.42.4.1	game	157
7.42.4.2	sender	157
7.43	Io23.battleship.online.network.messages.JoinGameResponseMessage Class Reference	158
7.43.1	Detailed Description	158
7.43.2	Constructor & Destructor Documentation	159
7.43.2.1	JoinGameResponseMessage(Boolean isOk, Profile sender, Game game)	159

7.43.3	Member Function Documentation	159
7.43.3.1	getType()	159
7.43.3.2	process(IDataCom IData, InetAddress senderAddress)	159
7.43.4	Member Data Documentation	160
7.43.4.1	game	160
7.43.4.2	isOk	160
7.43.4.3	sender	160
7.44	guiMain.controller.LoginController Class Reference	160
7.44.1	Member Function Documentation	160
7.44.1.1	initialize(URL location, ResourceBundle resources)	160
7.44.1.2	login(ActionEvent event)	160
7.44.1.3	openCreateAccountWindow(ActionEvent event)	161
7.44.1.4	setMainController(GuiMainController c)	161
7.44.2	Member Data Documentation	161
7.44.2.1	createAccount	161
7.44.2.2	errorMessage	161
7.44.2.3	loginButton	161
7.44.2.4	mainController	161
7.44.2.5	userLogin	161
7.44.2.6	userPassword	161
7.45	Io23.battleship.MainApp Class Reference	161
7.45.1	Member Function Documentation	162
7.45.1.1	main(String[] args)	162
7.45.1.2	start(Stage stage)	162
7.46	guiMain.controller.menuController Class Reference	162
7.46.1	Detailed Description	164
7.46.2	Member Function Documentation	164
7.46.2.1	addGame(Game game)	164
7.46.2.2	addUser(User user)	164
7.46.2.3	disconnection()	164

7.46.2.4	init()	164
7.46.2.5	initGamesList()	165
7.46.2.6	initialize(URL location, ResourceBundle resources)	165
7.46.2.7	initUserList()	165
7.46.2.8	joinGame(Game game)	165
7.46.2.9	lookGame(Game game)	165
7.46.2.10	lookUser(User user)	165
7.46.2.11	openChangeProfileWindow(ActionEvent event)	165
7.46.2.12	openCreateGameWindow(ActionEvent event)	166
7.46.2.13	option()	166
7.46.2.14	refresh()	166
7.46.2.15	removeGame(Game removedGame)	166
7.46.2.16	removeUser(User user)	166
7.46.2.17	setImage()	167
7.46.2.18	setMainController(GuiMainController c)	167
7.46.2.19	updateGameStatus(Game game)	167
7.46.3	Member Data Documentation	167
7.46.3.1	avatarImage	167
7.46.3.2	gamesView	167
7.46.3.3	mainController	167
7.46.3.4	modifyProfileButton	167
7.46.3.5	optionButton	167
7.46.3.6	playersView	167
7.46.3.7	refreshButton	167
7.46.3.8	warningLabel	167
7.47	Io23.battleship.online.network.messages.Message Class Reference	168
7.47.1	Detailed Description	168
7.47.2	Member Function Documentation	169
7.47.2.1	getType()	169
7.47.2.2	process(IDataCom IData, InetAddress senderAddress)	169

7.47.3	Member Data Documentation	169
7.47.3.1	type	169
7.48	structData.MessageType Enum Reference	169
7.48.1	Detailed Description	170
7.48.2	Member Data Documentation	170
7.48.2.1	ERROR	170
7.48.2.2	NOTIFICATION	170
7.49	Io23.battleship.online.network.NetworkController Class Reference	170
7.49.1	Detailed Description	171
7.49.2	Constructor & Destructor Documentation	172
7.49.2.1	NetworkController()	172
7.49.3	Member Function Documentation	172
7.49.3.1	addToNetwork(User sender, InetAddress senderAddress, Game game)	172
7.49.3.2	addUserToNetwork(User user, InetAddress senderAddress)	172
7.49.3.3	clearNetwork()	172
7.49.3.4	closeListener()	172
7.49.3.5	filterKnownIPAddressesToNotify(List< InetAddress > iPAddressesTable)	173
7.49.3.6	filterUnknownIPAddresses(List< InetAddress > iPAddressesTable)	173
7.49.3.7	getAddressForUser(User user)	173
7.49.3.8	getCOMInterface()	173
7.49.3.9	getConnectedUsers()	174
7.49.3.10	getInstance()	174
7.49.3.11	getIPTable()	174
7.49.3.12	getPort()	174
7.49.3.13	launchServer()	174
7.49.3.14	removeFromNetwork(User user, Game game)	174
7.49.3.15	removeUnreachableHost(InetAddress address)	174
7.49.3.16	sendMessage(Message message, InetAddress destinationIpAddress)	175
7.49.3.17	setDataInterface(IDataCom IData)	176
7.49.4	Member Data Documentation	176

7.49.4.1	dataInterface	176
7.49.4.2	instance	176
7.49.4.3	networkInterface	176
7.49.4.4	networkServer	176
7.49.4.5	networkState	176
7.49.4.6	port	176
7.50	Io23.battleship.online.network.NetworkListener Class Reference	176
7.50.1	Detailed Description	177
7.50.2	Constructor & Destructor Documentation	178
7.50.2.1	NetworkListener(NetworkServer server, ServerSocket socket)	178
7.50.3	Member Function Documentation	178
7.50.3.1	closeSocket()	178
7.50.3.2	getIsRunning()	178
7.50.3.3	getServerSocketIPAddress()	178
7.50.3.4	read()	179
7.50.3.5	run()	179
7.50.3.6	setDataInterface(IDataCom IData)	179
7.50.3.7	setIsRunning(boolean newValue)	179
7.50.4	Member Data Documentation	179
7.50.4.1	dataInterface	179
7.50.4.2	isRunning	179
7.50.4.3	reader	179
7.50.4.4	server	179
7.50.4.5	serverSocket	179
7.51	Io23.battleship.online.network.NetworkModuleInterface Class Reference	180
7.51.1	Detailed Description	181
7.51.2	Constructor & Destructor Documentation	181
7.51.2.1	NetworkModuleInterface(NetworkController cont)	181
7.51.3	Member Function Documentation	181
7.51.3.1	askDisconnection()	182

7.51.3.2	changeStatusGame(Game game)	182
7.51.3.3	clearNetwork()	182
7.51.3.4	coordinates(Player destPlayer, Shot resultShot, Game game, Boat boat)	182
7.51.3.5	gameQuitSpectator(User spec, Game game)	182
7.51.3.6	getInfoGameForSpectator(Player player, User spec)	183
7.51.3.7	getProfile(User userRequested)	183
7.51.3.8	joinGame(Game g)	183
7.51.3.9	notifyGameWon()	183
7.51.3.10	notifyJoinGameResponse(boolean isOk, Profile user, Game game)	183
7.51.3.11	notifyNewGame(Game g)	184
7.51.3.12	notifyReady(User user, Player playerToNotify)	184
7.51.3.13	removeGame(Game game)	184
7.51.3.14	searchForPlayers()	184
7.51.3.15	sendChatMessage(ChatMessage chatMessage, Game g)	185
7.51.3.16	sendInfoGameForSpectator(Game game, User spec)	185
7.51.3.17	sendNewSpectator(User u, Player p, HashSet< User > listSpectators)	185
7.51.3.18	sendShot(Player player, Game game, Shot shot)	185
7.51.3.19	setDataInterface(IDataCom IData)	186
7.51.4	Member Data Documentation	186
7.51.4.1	controller	186
7.51.4.2	dataInterface	186
7.52	Io23.battleship.online.network.NetworkSender Class Reference	186
7.52.1	Detailed Description	187
7.52.2	Constructor & Destructor Documentation	187
7.52.2.1	NetworkSender(InetAddress host, int port, Message message)	187
7.52.3	Member Function Documentation	188
7.52.3.1	run()	188
7.52.4	Member Data Documentation	188
7.52.4.1	controller	188
7.52.4.2	host	188

7.52.4.3	message	188
7.52.4.4	port	188
7.52.4.5	sock	188
7.52.4.6	writer	188
7.53	Io23.battleship.online.network.NetworkServer Class Reference	188
7.53.1	Detailed Description	189
7.53.2	Constructor & Destructor Documentation	189
7.53.2.1	NetworkServer(NetworkController networkController, int port)	189
7.53.3	Member Function Documentation	190
7.53.3.1	close()	190
7.53.3.2	getIpAddress()	190
7.53.3.3	isOpenned()	190
7.53.3.4	open()	190
7.53.3.5	setDataInterface(IDataCom IData)	190
7.53.4	Member Data Documentation	190
7.53.4.1	address	190
7.53.4.2	backlog	190
7.53.4.3	dataInterface	191
7.53.4.4	listener	191
7.53.4.5	networkController	191
7.53.4.6	port	191
7.54	Io23.battleship.online.network.messages.NotifyReadyMessage Class Reference	191
7.54.1	Detailed Description	192
7.54.2	Constructor & Destructor Documentation	192
7.54.2.1	NotifyReadyMessage(User sender, User destUser)	192
7.54.3	Member Function Documentation	192
7.54.3.1	getType()	192
7.54.3.2	process(IDataCom IData, InetAddress senderAddress)	192
7.54.4	Member Data Documentation	193
7.54.4.1	destUser	193

7.54.4.2	sender	193
7.55	guiTable.controllers.ObservationPhase Class Reference	193
7.55.1	Member Function Documentation	195
7.55.1.1	displayShot(Shot shot, int player)	195
7.55.1.2	exitGame(ActionEvent event)	195
7.55.1.3	getChatPane()	195
7.55.1.4	getLblj1()	195
7.55.1.5	getLblj2()	195
7.55.1.6	init()	195
7.55.1.7	logMsg(String msg)	195
7.55.1.8	logYesNoMsg(String msg)	196
7.55.1.9	noClicked(ActionEvent event)	196
7.55.1.10	placeShotTo(Shot shot, GridPane grid)	196
7.55.1.11	setLblj1(String lblj1)	196
7.55.1.12	setLblj2(String lblj2)	196
7.55.1.13	setTableController(GuiTableController c)	196
7.55.1.14	setTurn(Boolean t)	197
7.55.1.15	showVictory(int player)	197
7.55.1.16	sunkABoat(GridPane grid, Boat boat)	197
7.55.1.17	sunkPlayerBoat(int i, Boat boat)	197
7.55.1.18	yesClicked(ActionEvent event)	197
7.55.2	Member Data Documentation	197
7.55.2.1	anchorPane	197
7.55.2.2	chatPane	197
7.55.2.3	chatPane1	197
7.55.2.4	EXIT_GAME_MSG	198
7.55.2.5	exitButton	198
7.55.2.6	gameState	198
7.55.2.7	lblj1	198
7.55.2.8	lblj2	198

7.55.2.9	messageContainer	198
7.55.2.10	messageTextContainer	198
7.55.2.11	MY_TURN_MSG	198
7.55.2.12	noButton	198
7.55.2.13	OTHER_TURN_MSG	198
7.55.2.14	STYLE_MY_TURN	198
7.55.2.15	STYLE_OTHER_TURN	198
7.55.2.16	table1	198
7.55.2.17	table2	198
7.55.2.18	tableController	198
7.55.2.19	tablePlayer	198
7.55.2.20	timerLabel	198
7.55.2.21	turn	198
7.55.2.22	VICTORY_P1_MSG	198
7.55.2.23	VICTORY_P2_MSG	198
7.55.2.24	waitExit	198
7.55.2.25	waitMsg	198
7.55.2.26	yesButton	198
7.56	guiTable.controllers.ObserverPhaseController Class Reference	199
7.56.1	Detailed Description	199
7.56.2	Member Function Documentation	199
7.56.2.1	displayShot(Shot shot, int player)	199
7.56.2.2	exitGame(ActionEvent event)	199
7.56.2.3	initialize(URL location, ResourceBundle resources)	199
7.56.2.4	noClicked(ActionEvent event)	199
7.56.2.5	setTurn(Boolean turn)	199
7.56.2.6	sunkPlayerBoat(int player, Boat boat)	200
7.56.3	Member Data Documentation	200
7.56.3.1	gameState	200
7.56.3.2	table1	200

7.56.3.3	table2	200
7.56.3.4	tablePlayer	200
7.57	guiTable.controllers.PlacementPhaseController Class Reference	200
7.57.1	Detailed Description	202
7.57.2	Member Function Documentation	202
7.57.2.1	activateBoat()	202
7.57.2.2	allBoatsArePlaced()	202
7.57.2.3	closeMsg()	202
7.57.2.4	desactiveBoat()	202
7.57.2.5	disableRotation()	203
7.57.2.6	draw(BoatDrawing boat, Integer collIndex, Integer rowIndex)	203
7.57.2.7	drawBoatsNewPosition()	204
7.57.2.8	drawRotation(BoatDrawing boat)	204
7.57.2.9	enableRotation()	204
7.57.2.10	getChatPane()	204
7.57.2.11	initBoatMap()	204
7.57.2.12	initialize(URL location, ResourceBundle resources)	204
7.57.2.13	isIsValidate()	205
7.57.2.14	logMsg(String msg, String...param)	205
7.57.2.15	MousePlaceBoat()	205
7.57.2.16	onValidate()	205
7.57.2.17	placeBoat(BoatDrawing myBoat)	205
7.57.2.18	playKeyEvent()	206
7.57.2.19	positionCorrect(BoatDrawing activeBoat)	206
7.57.2.20	reinitBoat(BoatDrawing boat)	206
7.57.2.21	setIsValidate(boolean isValidate)	206
7.57.2.22	setPlacementTime(Integer placementTime)	206
7.57.2.23	timeIsOver()	207
7.57.3	Member Data Documentation	207
7.57.3.1	activeBoat	207

7.57.3.2	anchorPane	207
7.57.3.3	boatMap	207
7.57.3.4	chatPane	207
7.57.3.5	EXPLAIN_PLACEMENT	207
7.57.3.6	isValidate	207
7.57.3.7	messageContainer	207
7.57.3.8	RANDOM_ROTATION	207
7.57.3.9	rotationIsValide	207
7.57.3.10	table	207
7.57.3.11	time	207
7.57.3.12	timeline	207
7.57.3.13	timePlacement	207
7.57.3.14	timerLabel	207
7.57.3.15	valider	207
7.58	structData.Player Class Reference	208
7.58.1	Detailed Description	209
7.58.2	Constructor & Destructor Documentation	209
7.58.2.1	Player(DataUser dUser)	209
7.58.2.2	Player(Profile p)	209
7.58.2.3	Player(Profile p, List< Boat > listBoatsData, HashSet< Shot > listShotsData)	209
7.58.3	Member Function Documentation	209
7.58.3.1	addBoat(Boat newBoat)	209
7.58.3.2	addShot(Shot newShot)	210
7.58.3.3	clonePlayer(Player playerClone)	210
7.58.3.4	compareProfileToPlayer(Profile p)	210
7.58.3.5	getListBoats()	210
7.58.3.6	getListShots()	211
7.58.3.7	getProfile()	211
7.58.3.8	isReady()	211
7.58.3.9	setListBoats(List< Boat > listBoatsData)	211

7.58.3.10	setListShots(HashSet< Shot > listShotsData)	211
7.58.3.11	setProfile(Profile p)	211
7.58.3.12	setReady(boolean value)	212
7.58.4	Member Data Documentation	212
7.58.4.1	listBoats	212
7.58.4.2	listShots	212
7.58.4.3	profile	212
7.58.4.4	ready	212
7.58.4.5	serialVersionUID	212
7.59	guiMain.PlayerCell Class Reference	212
7.59.1	Detailed Description	213
7.59.2	Constructor & Destructor Documentation	213
7.59.2.1	PlayerCell(menuController c)	213
7.59.3	Member Function Documentation	213
7.59.3.1	updateItem(final User user, boolean empty)	213
7.59.4	Member Data Documentation	213
7.59.4.1	controller	213
7.59.4.2	hbox	214
7.59.4.3	lookButton	214
7.59.4.4	lookButtonBox	214
7.59.4.5	name	214
7.60	structData.Position Class Reference	214
7.60.1	Detailed Description	215
7.60.2	Constructor & Destructor Documentation	215
7.60.2.1	Position()	215
7.60.2.2	Position(int newX, int newY, Boolean newTouched)	215
7.60.2.3	Position(Position p)	215
7.60.3	Member Function Documentation	215
7.60.3.1	clone(Position p)	215
7.60.3.2	getTouched()	216

7.60.3.3	getX()	216
7.60.3.4	getY()	216
7.60.3.5	setTouched(Boolean touched)	216
7.60.3.6	setX(int x)	216
7.60.3.7	setY(int y)	217
7.60.4	Member Data Documentation	217
7.60.4.1	serialVersionUID	217
7.60.4.2	touched	217
7.60.4.3	x	217
7.60.4.4	y	217
7.61	guiMain.controller.ProfilController Class Reference	217
7.61.1	Detailed Description	218
7.61.2	Member Function Documentation	218
7.61.2.1	backToTheMenu(ActionEvent event)	218
7.61.2.2	init(User user)	218
7.61.2.3	initialize(URL location, ResourceBundle resources)	219
7.61.2.4	setMainController(GuiMainController c)	219
7.61.2.5	setProfil(Profile profile)	219
7.61.3	Member Data Documentation	219
7.61.3.1	backButton	219
7.61.3.2	birthdate	219
7.61.3.3	firstName	219
7.61.3.4	lastName	219
7.61.3.5	mainController	219
7.61.3.6	nameTitle	219
7.61.3.7	numberOfGame	219
7.61.3.8	numberOfGameLost	219
7.61.3.9	numberOfGameWon	220
7.61.3.10	userAvatar	220
7.61.3.11	userName	220

7.62 structData.Profile Class Reference	220
7.62.1 Detailed Description	221
7.62.2 Constructor & Destructor Documentation	221
7.62.2.1 Profile(DataUser dUser)	221
7.62.2.2 Profile(DataUser dUser, String pathToImage, String newLastname, String new↵ Name, Date newBirthdate)	222
7.62.2.3 Profile(DataUser dUser, ImageIcon image, String newLastname, String new↵ Name, Date newBirthdate)	222
7.62.3 Member Function Documentation	222
7.62.3.1 cleanUp(File file)	222
7.62.3.2 clone(Profile p)	222
7.62.3.3 getAvatar()	223
7.62.3.4 getBirthdate()	223
7.62.3.5 getGamesAborted()	223
7.62.3.6 getGamesLost()	223
7.62.3.7 getGamesPlayed()	223
7.62.3.8 getGamesWon()	224
7.62.3.9 getImage()	224
7.62.3.10 getLastname()	224
7.62.3.11 getName()	224
7.62.3.12 saveeditedProfile()	224
7.62.3.13 saveProfile()	224
7.62.3.14 setAvatar(ImageIcon i)	224
7.62.3.15 setAvatar(String path)	225
7.62.3.16 setBirthdate(Date b)	225
7.62.3.17 setGamesAborted(int nb)	225
7.62.3.18 setGamesLost(int nb)	225
7.62.3.19 setGamesPlayed(int nb)	225
7.62.3.20 setGamesWon(int nb)	226
7.62.3.21 setLastname(String ln)	226
7.62.3.22 setName(String n)	226

7.62.4	Member Data Documentation	226
7.62.4.1	avatar	226
7.62.4.2	birthdate	226
7.62.4.3	gamesAborted	226
7.62.4.4	gamesLost	226
7.62.4.5	gamesPlayed	226
7.62.4.6	gamesWon	226
7.62.4.7	lastname	226
7.62.4.8	name	226
7.63	guiTable.controllers.ProfileController Class Reference	227
7.63.1	Member Function Documentation	227
7.63.1.1	init()	227
7.64	Io23.battleship.online.network.messages.ProfileRequestAnswerMessage Class Reference	227
7.64.1	Detailed Description	228
7.64.2	Constructor & Destructor Documentation	228
7.64.2.1	ProfileRequestAnswerMessage(Profile profile)	228
7.64.3	Member Function Documentation	228
7.64.3.1	getType()	228
7.64.3.2	process(IDataCom IData, InetAddress senderAddress)	228
7.64.4	Member Data Documentation	229
7.64.4.1	profile	229
7.64.4.2	user	229
7.65	Io23.battleship.online.network.messages.SendInfoGameForSpectatorMessage Class Reference	229
7.65.1	Detailed Description	230
7.65.2	Constructor & Destructor Documentation	230
7.65.2.1	SendInfoGameForSpectatorMessage(Game g, User s)	230
7.65.3	Member Function Documentation	230
7.65.3.1	getType()	230
7.65.3.2	process(IDataCom IData, InetAddress senderAddress)	230
7.65.4	Member Data Documentation	231

7.65.4.1	game	231
7.65.4.2	spec	231
7.66	Io23.battleship.online.network.messages.SendNewSpectatorMessage Class Reference	231
7.66.1	Detailed Description	232
7.66.2	Constructor & Destructor Documentation	232
7.66.2.1	SendNewSpectatorMessage(User s)	232
7.66.3	Member Function Documentation	232
7.66.3.1	getType()	232
7.66.3.2	process(IDataCom IData, InetAddress senderAddress)	233
7.66.4	Member Data Documentation	233
7.66.4.1	spec	233
7.67	Io23.battleship.online.network.messages.SendTextMessage Class Reference	233
7.67.1	Detailed Description	234
7.67.2	Constructor & Destructor Documentation	234
7.67.2.1	SendTextMessage(ChatMessage m)	234
7.67.3	Member Function Documentation	234
7.67.3.1	getType()	234
7.67.3.2	process(IDataCom IData, InetAddress senderAddress)	235
7.67.4	Member Data Documentation	235
7.67.4.1	message	235
7.68	structData.Shot Class Reference	235
7.68.1	Detailed Description	236
7.68.2	Constructor & Destructor Documentation	236
7.68.2.1	Shot(Position p)	236
7.68.2.2	Shot(Position p, Date nTime)	236
7.68.3	Member Function Documentation	236
7.68.3.1	getTime()	236
7.68.3.2	setTime(Date nTime)	236
7.68.4	Member Data Documentation	237
7.68.4.1	time	237

7.69	Io23.battleship.online.network.messages.ShotNotificationMessage Class Reference	237
7.69.1	Detailed Description	238
7.69.2	Constructor & Destructor Documentation	238
7.69.2.1	ShotNotificationMessage(Shot s)	238
7.69.3	Member Function Documentation	238
7.69.3.1	getType()	238
7.69.3.2	process(IDataCom IData, InetAddress senderAddress)	238
7.69.4	Member Data Documentation	239
7.69.4.1	shot	239
7.70	Io23.battleship.online.network.messages.ShotNotificationResultForSpectatorMessage Class Reference	239
7.70.1	Detailed Description	240
7.70.2	Constructor & Destructor Documentation	240
7.70.2.1	ShotNotificationResultForSpectatorMessage(Player p, Shot s, Boat b)	240
7.70.3	Member Function Documentation	241
7.70.3.1	getType()	241
7.70.3.2	process(IDataCom IData, InetAddress senderAddress)	241
7.70.4	Member Data Documentation	241
7.70.4.1	boat	241
7.70.4.2	player	241
7.70.4.3	shot	241
7.71	Io23.battleship.online.network.messages.ShotNotificationResultMessage Class Reference	242
7.71.1	Detailed Description	242
7.71.2	Constructor & Destructor Documentation	242
7.71.2.1	ShotNotificationResultMessage(Shot s, Boat b)	242
7.71.3	Member Function Documentation	243
7.71.3.1	getType()	243
7.71.3.2	process(IDataCom IData, InetAddress senderAddress)	243
7.71.4	Member Data Documentation	243
7.71.4.1	boat	243
7.71.4.2	shot	243

7.72	guiMain.controller.SignupController Class Reference	244
7.72.1	Detailed Description	244
7.72.2	Member Function Documentation	245
7.72.2.1	addConnectionPoint()	245
7.72.2.2	backToConnectionWindow()	245
7.72.2.3	chooseAvatar(ActionEvent event)	245
7.72.2.4	createAccount()	245
7.72.2.5	init()	245
7.72.2.6	openFile(File file)	245
7.72.2.7	setMainController(GuiMainController c)	245
7.72.3	Member Data Documentation	245
7.72.3.1	addConnectionButton	245
7.72.3.2	avatarImage	246
7.72.3.3	avatarPath	246
7.72.3.4	birthDatePicker	246
7.72.3.5	chooseAvatarButton	246
7.72.3.6	connectionPageButton	246
7.72.3.7	createButton	246
7.72.3.8	firstNameTextField	246
7.72.3.9	idTextField	246
7.72.3.10	lastNameTextField	246
7.72.3.11	mainController	246
7.72.3.12	messageLabel	246
7.72.3.13	passwordTextField	246
7.72.3.14	usernameTextField	246
7.73	structData.StatusGame Enum Reference	246
7.73.1	Detailed Description	247
7.73.2	Member Data Documentation	247
7.73.2.1	BOATPHASE	247
7.73.2.2	FINISHED	247

7.73.2.3	PLAYER1READY	247
7.73.2.4	PLAYER2READY	247
7.73.2.5	PLAYING	247
7.73.2.6	WAITINGBOT	247
7.73.2.7	WAITINGPLAYER	247
7.74	guiTable.CaseDrawing.Type Enum Reference	247
7.74.1	Member Data Documentation	247
7.74.1.1	BOAT	247
7.74.1.2	MISSED	247
7.74.1.3	SHOT	247
7.74.1.4	SUNK_BOAT	247
7.74.1.5	TOUCHED	247
7.75	Io23.battleship.online.network.messages.UpdateGameMessage Class Reference	248
7.75.1	Detailed Description	248
7.75.2	Constructor & Destructor Documentation	248
7.75.2.1	UpdateGameMessage(Game game)	248
7.75.3	Member Function Documentation	249
7.75.3.1	getType()	249
7.75.3.2	process(IDataCom IData, InetAddress senderAddress)	249
7.75.4	Member Data Documentation	249
7.75.4.1	gameUpdate	249
7.76	structData.User Class Reference	250
7.76.1	Detailed Description	251
7.76.2	Constructor & Destructor Documentation	251
7.76.2.1	User(String loginData, String usernameData)	251
7.76.2.2	User(User u)	251
7.76.3	Member Function Documentation	251
7.76.3.1	cloneUser(User userClone)	251
7.76.3.2	getIdUser()	252
7.76.3.3	getIPs()	252

7.76.3.4	getLogin()	252
7.76.3.5	getPort()	252
7.76.3.6	getUsername()	252
7.76.3.7	setIdUser(UUID idUserData)	252
7.76.3.8	setIPs(HashSet iPsData)	253
7.76.3.9	setLogin(String loginData)	253
7.76.3.10	setPort(int p)	253
7.76.3.11	setUsername(String usernameData)	253
7.76.4	Member Data Documentation	253
7.76.4.1	DEFAULT_PORT	253
7.76.4.2	idUser	253
7.76.4.3	iPs	254
7.76.4.4	login	254
7.76.4.5	port	254
7.76.4.6	serialVersionUID	254
7.76.4.7	username	254
7.77	guiMain.controller.WaitingRoomController Class Reference	254
7.77.1	Member Function Documentation	254
7.77.1.1	closeWindow()	254
7.77.1.2	initData(Game game)	254
7.77.1.3	initialize(URL location, ResourceBundle resources)	255
7.77.1.4	setStage(Stage s)	255
7.77.2	Member Data Documentation	255
7.77.2.1	chatAutorisation	255
7.77.2.2	gameName	255
7.77.2.3	gameType	255
7.77.2.4	reflectionTime	255
7.77.2.5	spectatorAutorisation	255
7.77.2.6	stage	255

8	File Documentation	257
8.1	README.md File Reference	257
8.2	src/main/java/data/CDataCom.java File Reference	257
8.3	src/main/java/data/CDataMain.java File Reference	257
8.4	src/main/java/data/CDataTable.java File Reference	257
8.5	src/main/java/data/DataController.java File Reference	258
8.6	src/main/java/guiMain/controller/ChangeProfileController.java File Reference	258
8.7	src/main/java/guiMain/controller/CreateGameController.java File Reference	258
8.8	src/main/java/guiMain/controller/IpConfigController.java File Reference	258
8.9	src/main/java/guiMain/controller/LoginController.java File Reference	259
8.10	src/main/java/guiMain/controller/menuController.java File Reference	259
8.11	src/main/java/guiMain/controller/ProfilController.java File Reference	259
8.12	src/main/java/guiMain/controller/SignupController.java File Reference	259
8.13	src/main/java/guiMain/controller/WaitingRoomController.java File Reference	260
8.14	src/main/java/guiMain/GameCell.java File Reference	260
8.15	src/main/java/guiMain/GuiMainController.java File Reference	260
8.16	src/main/java/guiMain/GuiMainInterface.java File Reference	260
8.17	src/main/java/guiMain/IpCell.java File Reference	261
8.18	src/main/java/guiMain/PlayerCell.java File Reference	261
8.19	src/main/java/guiTable/BoatDrawing.java File Reference	261
8.20	src/main/java/guiTable/CaseDrawing.java File Reference	261
8.21	src/main/java/guiTable/controllers/BaseController.java File Reference	262
8.22	src/main/java/guiTable/controllers/BelgianPlacementPhaseController.java File Reference	262
8.23	src/main/java/guiTable/controllers/ChatController.java File Reference	262
8.24	src/main/java/guiTable/controllers/ClassicPlacementPhaseController.java File Reference	262
8.25	src/main/java/guiTable/controllers/gameInterface.java File Reference	263
8.26	src/main/java/guiTable/controllers/GamePhaseController.java File Reference	263
8.27	src/main/java/guiTable/controllers/GuiTableController.java File Reference	263
8.28	src/main/java/guiTable/controllers/ObservationPhase.java File Reference	263
8.29	src/main/java/guiTable/controllers/ObserverPhaseController.java File Reference	264

8.30	src/main/java/guiTable/controllers/PlacementPhaseController.java File Reference	264
8.31	src/main/java/guiTable/controllers/ProfileController.java File Reference	264
8.32	src/main/java/guiTable/GuiTableInterface.java File Reference	264
8.33	src/main/java/interfacesData/IDataCom.java File Reference	265
8.34	src/main/java/interfacesData/IDataMain.java File Reference	265
8.35	src/main/java/interfacesData/IDataTable.java File Reference	265
8.36	src/main/java/lo23/battleship/MainApp.java File Reference	265
8.37	src/main/java/lo23/battleship/online/network/COMInterface.java File Reference	266
8.38	src/main/java/lo23/battleship/online/network/messages/ConnectionEstablishedMessage.java File Reference	266
8.39	src/main/java/lo23/battleship/online/network/messages/ConnectionRequestMessage.java File Reference	266
8.40	src/main/java/lo23/battleship/online/network/messages/CreatedGameNotificationMessage.java File Reference	267
8.41	src/main/java/lo23/battleship/online/network/messages/DisconnectionMessage.java File Reference	267
8.42	src/main/java/lo23/battleship/online/network/messages/GameQuitMessage.java File Reference	267
8.43	src/main/java/lo23/battleship/online/network/messages/GameQuitSpectatorMessage.java File Reference	268
8.44	src/main/java/lo23/battleship/online/network/messages/GameWonMessage.java File Reference	268
8.45	src/main/java/lo23/battleship/online/network/messages/GameWonMessageToSpectator.java File Reference	268
8.46	src/main/java/lo23/battleship/online/network/messages/GetInfoGameForSpectatorMessage.java File Reference	269
8.47	src/main/java/lo23/battleship/online/network/messages/GetProfileRequestMessage.java File Reference	269
8.48	src/main/java/lo23/battleship/online/network/messages/JoinGameRequestMessage.java File Reference	269
8.49	src/main/java/lo23/battleship/online/network/messages/JoinGameResponseMessage.java File Reference	270
8.50	src/main/java/lo23/battleship/online/network/messages/Message.java File Reference	270
8.51	src/main/java/lo23/battleship/online/network/messages/NotifyReadyMessage.java File Reference	270
8.52	src/main/java/lo23/battleship/online/network/messages/package-info.java File Reference	271
8.53	src/main/java/lo23/battleship/online/network/package-info.java File Reference	271
8.54	src/main/java/lo23/battleship/online/network/messages/ProfileRequestAnswerMessage.java File Reference	271

8.55	src/main/java/lo23/battleship/online/network/messages/SendInfoGameForSpectatorMessage.java File Reference	271
8.56	src/main/java/lo23/battleship/online/network/messages/SendNewSpectatorMessage.java File Ref- erence	272
8.57	src/main/java/lo23/battleship/online/network/messages/SendTextMessage.java File Reference . . .	272
8.58	src/main/java/lo23/battleship/online/network/messages/ShotNotificationMessage.java File Reference	272
8.59	src/main/java/lo23/battleship/online/network/messages/ShotNotificationResultForSpectator↵ Message.java File Reference	273
8.60	src/main/java/lo23/battleship/online/network/messages/ShotNotificationResultMessage.java File Reference	273
8.61	src/main/java/lo23/battleship/online/network/messages/UpdateGameMessage.java File Reference .	273
8.62	src/main/java/lo23/battleship/online/network/NetworkController.java File Reference	274
8.63	src/main/java/lo23/battleship/online/network/NetworkListener.java File Reference	274
8.64	src/main/java/lo23/battleship/online/network/NetworkModuleInterface.java File Reference	274
8.65	src/main/java/lo23/battleship/online/network/NetworkSender.java File Reference	274
8.66	src/main/java/lo23/battleship/online/network/NetworkServer.java File Reference	275
8.67	src/main/java/structData/Boat.java File Reference	275
8.68	src/main/java/structData/BoatType.java File Reference	275
8.69	src/main/java/structData/ChatMessage.java File Reference	276
8.70	src/main/java/structData/ContactGroup.java File Reference	276
8.71	src/main/java/structData/DataUser.java File Reference	276
8.72	src/main/java/structData/Game.java File Reference	276
8.73	src/main/java/structData/MessageType.java File Reference	277
8.74	src/main/java/structData/Player.java File Reference	277
8.75	src/main/java/structData/Position.java File Reference	277
8.76	src/main/java/structData/Profile.java File Reference	277
8.77	src/main/java/structData/Shot.java File Reference	278
8.78	src/main/java/structData/StatusGame.java File Reference	278
8.79	src/main/java/structData/User.java File Reference	278

Chapter 1

L023-Battleship-Online

Software Project Management - Online Battleship Game

documentation

to generate documenation run :

```
1 doxygen Doxyfile
```

It will create an html documentation available at

```
1 ./doc/html/index.html
```

and a latex documentation which need to be compiled :

```
1 cd ./doc/latex/  
2 make
```

then, you can find output in pdf here :

```
1 ./doc/latex/refman.pdf
```


Chapter 2

Namespace Index

2.1 Packages

Here are the packages with brief descriptions (if available):

data	13
guiMain	13
guiMain.controller	14
guiTable	14
guiTable.controllers	14
interfacesData	15
lo23	15
lo23.battleship	15
lo23.battleship.online	15
lo23.battleship.online.network	
Created by xzirva on 17/10/17	16
lo23.battleship.online.network.messages	
Created by xzirva on 17/10/17	16
structData	18

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

guiTable.controllers.BaseController	19
guiTable.controllers.gameInterface	100
guiTable.controllers.GamePhaseController	103
guiTable.controllers.ObserverPhaseController	199
guiTable.controllers.ObservationPhase	193
guiTable.controllers.PlacementPhaseController	200
guiTable.controllers.BelgianPlacementPhaseController	21
guiTable.controllers.ClassicPlacementPhaseController	57
guiTable.BoatDrawing	26
structData.BoatType	30
guiTable.controllers.ChatController	53
lo23.battleship.online.network.COMInterface	58
lo23.battleship.online.network.NetworkModuleInterface	180
data.DataController	74
guiMain.GuiMainInterface	126
guiMain.GuiMainController	119
guiTable.GuiTableInterface	133
guiTable.controllers.GuiTableController	128
interfacesData.IDataCom	137
data.CDataCom	32
interfacesData.IDataMain	145
data.CDataMain	41
interfacesData.IDataTable	150
data.CDataTable	46
guiMain.controller.IpConfigController	154
structData.MessageType	169
lo23.battleship.online.network.NetworkController	170
lo23.battleship.online.network.NetworkServer	188
guiTable.controllers.ProfileController	227
guiMain.controller.SignupController	244
structData.StatusGame	246
Thread	
lo23.battleship.online.network.NetworkListener	176

lo23.battleship.online.network.NetworkSender	186
guiTable.CaseDrawing.Type	247
Application	
lo23.battleship.MainApp	161
Initializable	
guiMain.controller.ChangeProfileController	50
guiMain.controller.CreateGameController	71
guiMain.controller.LoginController	160
guiMain.controller.menuController	162
guiMain.controller.ProfilController	217
guiMain.controller.WaitingRoomController	254
guiTable.controllers.gameInterface	100
guiTable.controllers.GamePhaseController	103
guiTable.controllers.ObserverPhaseController	199
guiTable.controllers.PlacementPhaseController	200
Label	
guiTable.CaseDrawing	31
ListCell	
guiMain.GameCell	98
guiMain.IpCell	152
guiMain.PlayerCell	212
Serializable	
lo23.battleship.online.network.messages.Message	168
lo23.battleship.online.network.messages.ConnectionEstablishedMessage	64
lo23.battleship.online.network.messages.ConnectionRequestMessage	66
lo23.battleship.online.network.messages.CreatedGameNotificationMessage	69
lo23.battleship.online.network.messages.DisconnectionMessage	88
lo23.battleship.online.network.messages.GameQuitMessage	108
lo23.battleship.online.network.messages.GameQuitSpectatorMessage	109
lo23.battleship.online.network.messages.GameWonMessage	111
lo23.battleship.online.network.messages.GameWonMessageToSpectator	113
lo23.battleship.online.network.messages.GetInfoGameForSpectatorMessage	115
lo23.battleship.online.network.messages.GetProfileRequestMessage	117
lo23.battleship.online.network.messages.JoinGameRequestMessage	156
lo23.battleship.online.network.messages.JoinGameResponseMessage	158
lo23.battleship.online.network.messages.NotifyReadyMessage	191
lo23.battleship.online.network.messages.ProfileRequestAnswerMessage	227
lo23.battleship.online.network.messages.SendInfoGameForSpectatorMessage	229
lo23.battleship.online.network.messages.SendNewSpectatorMessage	231
lo23.battleship.online.network.messages.SendTextMessage	233
lo23.battleship.online.network.messages.ShotNotificationMessage	237
lo23.battleship.online.network.messages.ShotNotificationResultForSpectatorMessage	239
lo23.battleship.online.network.messages.ShotNotificationResultMessage	242
lo23.battleship.online.network.messages.UpdateGameMessage	248
structData.Boat	22
structData.ChatMessage	54
structData.ContactGroup	68
structData.Game	90
structData.Player	208
structData.Position	214
structData.Shot	235
structData.User	250
structData.DataUser	85
structData.Profile	220

Chapter 4

Class Index

4.1 Class List

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

guiTable.controllers.BaseController	19
Every class controller will herit from BaseController	
guiTable.controllers.BelgianPlacementPhaseController	21
Implementation of placement controller for belgian phase	
structData.Boat	22
Boat is the class which represents a ship of a player	
guiTable.BoatDrawing	26
Boat drawing : class to draw a boat	
structData.BoatType	30
BoatType is an emumerate of all the types of boats and numbur of cases associated	
guiTable.CaseDrawing	31
CaseDrawing : class to draw a case	
data.CDataCom	32
CDataCom : interface of Data for Com	
data.CDataMain	41
CDataCom : interface of Data for IHM Main	
data.CDataTable	46
Data's interface for Table	
guiMain.controller.ChangeProfileController	50
guiTable.controllers.ChatController	53
structData.ChatMessage	54
ChatMessage is a class for chat messages	
guiTable.controllers.ClassicPlacementPhaseController	57
Implementation of placement controller for classic phase	
lo23.battleship.online.network.COMInterface	58
The COMInterface compiles the services offered by the network package	
lo23.battleship.online.network.messages.ConnectionEstablishedMessage	64
This class implements the message which is sent when when a user disconnects from the appli- cation	
lo23.battleship.online.network.messages.ConnectionRequestMessage	66
Network message class, extends Message class	
structData.ContactGroup	68
ContactGroup is a class a contact groups	
lo23.battleship.online.network.messages.CreatedGameNotificationMessage	69
Network message class, extends Message class	

guiMain.controller.CreateGameController	71
data.DataController	
Data's controller	74
structData.DataUser	
DataUser class : inherits from User	85
lo23.battleship.online.network.messages.DisconnectionMessage	
This class implements the message which is sent when when a user disconnects from the application	88
structData.Game	
Game is a class for the user's game	90
guiMain.GameCell	
GameCell, descendant class ListCell<Game>, display the game name and offers the possibility to join or watch a game	98
guiTable.controllers.gameInterface	
GameInterface	100
guiTable.controllers.GamePhaseController	
GamePhaseController	103
lo23.battleship.online.network.messages.GameQuitMessage	
This class implements the message sent when a game is removed from the network (and the list of games) by his creator(user) or when the game is over	108
lo23.battleship.online.network.messages.GameQuitSpectatorMessage	
This class implements the message which is sent when a spectator quit a game he is watching	109
lo23.battleship.online.network.messages.GameWonMessage	
This class implements the message which is sent when a game is over with a victory of one of the players	111
lo23.battleship.online.network.messages.GameWonMessageToSpectator	
This class implements the message which is sent when a game is over with a victory of one of the players	113
lo23.battleship.online.network.messages.GetInfoGameForSpectatorMessage	
This class implements the message which is sent when when a user wants to watch a game	115
lo23.battleship.online.network.messages.GetProfileRequestMessage	
This class implements the message which is sent when when a user wants to see the profile(including statistics) of an other user	117
guiMain.GuiMainController	
This class implements the network controller	119
guiMain.GuiMainInterface	
GUI Main Interface	126
guiTable.controllers.GuiTableController	
GuiTableController	128
guiTable.GuiTableInterface	
Interface for the Ihm Table team	133
interfacesData.IDataCom	
Data's interface for Com	137
interfacesData.IDataMain	
Data's interface for Main	145
interfacesData.IDataTable	
Data's interface for IHM-Table	150
guiMain.IpCell	
IpCell, Display a list of Ips on a list that can be removed with a button	152
guiMain.controller.IpConfigController	
This class implements the controller of the Ip Configuration page	154
lo23.battleship.online.network.messages.JoinGameRequestMessage	
This class implements the message which is sent when when a user wants to join a game created by another user	156
lo23.battleship.online.network.messages.JoinGameResponseMessage	
This class implements the message which is sent when when a creator of a game adds a second player to the game	158
guiMain.controller.LoginController	160

lo23.battleship.MainApp	161
guiMain.controller.menuController	
This class call the lobby view and enable interaction between button and method	162
lo23.battleship.online.network.messages.Message	
This abstract class implements the core structure of a message sent on the network	168
structData.MessageType	
Enumerate for the types of messages (Class Message)	169
lo23.battleship.online.network.NetworkController	
This class implements the network controller	170
lo23.battleship.online.network.NetworkListener	
This class implements network listeners	176
lo23.battleship.online.network.NetworkModuleInterface	
This class implements the COMInterface interface and the different methods (services) it offers	180
lo23.battleship.online.network.NetworkSender	
This class implements network senders	186
lo23.battleship.online.network.NetworkServer	
This class implements the network server	188
lo23.battleship.online.network.messages.NotifyReadyMessage	
This class implements the message sent by the local user (player) to the other player when he is ready to play	191
guiTable.controllers.ObservationPhase	193
guiTable.controllers.ObserverPhaseController	
ObserverPhaseController	199
guiTable.controllers.PlacementPhaseController	
PlacementPhaseController	200
structData.Player	
Player is a class for players in a game	208
guiMain.PlayerCell	
PlayerCell, descendant class ListCell<User>, display the user name and offers the possibility watch his profil	212
structData.Position	
Position is a class of a position in a table	214
guiMain.controller.ProfilController	
GameCell, implement interface Initializable, display the user profil	217
structData.Profile	
Profile is a class for the user's profile	220
guiTable.controllers.ProfileController	227
lo23.battleship.online.network.messages.ProfileRequestAnswerMessage	
Network message class, extends Message class	227
lo23.battleship.online.network.messages.SendInfoGameForSpectatorMessage	
This class implements the message sent when the local user has been accepted as spectator	229
lo23.battleship.online.network.messages.SendNewSpectatorMessage	
This class implements the message sent when a user has been accepted as spectator by the creator of the game	231
lo23.battleship.online.network.messages.SendTextMessage	
This class implements textual messages sent/received by the local user to/from a distant user	233
structData.Shot	
Shot is a class of a shot of a player	235
lo23.battleship.online.network.messages.ShotNotificationMessage	
This class implements the message sent by the local user (player) when he shots	237
lo23.battleship.online.network.messages.ShotNotificationResultForSpectatorMessage	
This class implements the message sent by the local user (player) to the spectators with the result of the other player shot (missed, touched or sunk boat)	239
lo23.battleship.online.network.messages.ShotNotificationResultMessage	
This class implements the message sent by the local user (player) to the other player with the result of the other player's shot (missed, touched or sunk boat)	242
guiMain.controller.SignupController	
This class implements the controller of the SignUp page	244

structData.StatusGame	
StatusGame is an emumerate of all the status of games	246
guiTable.CaseDrawing.Type	247
lo23.battleship.online.network.messages.UpdateGameMessage	
This class implements the message sent by the local user to other users when a game (namely the status of the game) has changed	248
structData.User	
Class User	250
guiMain.controller.WaitingRoomController	254

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

src/main/java/data/CDDataCom.java	257
src/main/java/data/CDDataMain.java	257
src/main/java/data/CDDataTable.java	257
src/main/java/data/DataController.java	258
src/main/java/guiMain/GameCell.java	260
src/main/java/guiMain/GuiMainController.java	260
src/main/java/guiMain/GuiMainInterface.java	260
src/main/java/guiMain/IpCell.java	261
src/main/java/guiMain/PlayerCell.java	261
src/main/java/guiMain/controller/ChangeProfileController.java	258
src/main/java/guiMain/controller/CreateGameController.java	258
src/main/java/guiMain/controller/IpConfigController.java	258
src/main/java/guiMain/controller/LoginController.java	259
src/main/java/guiMain/controller/menuController.java	259
src/main/java/guiMain/controller/ProfilController.java	259
src/main/java/guiMain/controller/SignupController.java	259
src/main/java/guiMain/controller/WaitingRoomController.java	260
src/main/java/guiTable/BoatDrawing.java	261
src/main/java/guiTable/CaseDrawing.java	261
src/main/java/guiTable/GuiTableInterface.java	264
src/main/java/guiTable/controllers/BaseController.java	262
src/main/java/guiTable/controllers/BelgianPlacementPhaseController.java	262
src/main/java/guiTable/controllers/ChatController.java	262
src/main/java/guiTable/controllers/ClassicPlacementPhaseController.java	262
src/main/java/guiTable/controllers/gameInterface.java	263
src/main/java/guiTable/controllers/GamePhaseController.java	263
src/main/java/guiTable/controllers/GuiTableController.java	263
src/main/java/guiTable/controllers/ObservationPhase.java	263
src/main/java/guiTable/controllers/ObserverPhaseController.java	264
src/main/java/guiTable/controllers/PlacementPhaseController.java	264
src/main/java/guiTable/controllers/ProfileController.java	264
src/main/java/interfacesData/IDataCom.java	265
src/main/java/interfacesData/IDataMain.java	265
src/main/java/interfacesData/IDataTable.java	265
src/main/java/lo23/battleship/MainApp.java	265

src/main/java/lo23/battleship/online/network/COMInterface.java	266
src/main/java/lo23/battleship/online/network/NetworkController.java	274
src/main/java/lo23/battleship/online/network/NetworkListener.java	274
src/main/java/lo23/battleship/online/network/NetworkModuleInterface.java	274
src/main/java/lo23/battleship/online/network/NetworkSender.java	274
src/main/java/lo23/battleship/online/network/NetworkServer.java	275
src/main/java/lo23/battleship/online/network/package-info.java	271
src/main/java/lo23/battleship/online/network/messages/ConnectionEstablishedMessage.java	266
src/main/java/lo23/battleship/online/network/messages/ConnectionRequestMessage.java	266
src/main/java/lo23/battleship/online/network/messages/CreatedGameNotificationMessage.java	267
src/main/java/lo23/battleship/online/network/messages/DisconnectionMessage.java	267
src/main/java/lo23/battleship/online/network/messages/GameQuitMessage.java	267
src/main/java/lo23/battleship/online/network/messages/GameQuitSpectatorMessage.java	268
src/main/java/lo23/battleship/online/network/messages/GameWonMessage.java	268
src/main/java/lo23/battleship/online/network/messages/GameWonMessageToSpectator.java	268
src/main/java/lo23/battleship/online/network/messages/GetInfoGameForSpectatorMessage.java	269
src/main/java/lo23/battleship/online/network/messages/GetProfileRequestMessage.java	269
src/main/java/lo23/battleship/online/network/messages/JoinGameRequestMessage.java	269
src/main/java/lo23/battleship/online/network/messages/JoinGameResponseMessage.java	270
src/main/java/lo23/battleship/online/network/messages/Message.java	270
src/main/java/lo23/battleship/online/network/messages/NotifyReadyMessage.java	270
src/main/java/lo23/battleship/online/network/messages/package-info.java	271
src/main/java/lo23/battleship/online/network/messages/ProfileRequestAnswerMessage.java	271
src/main/java/lo23/battleship/online/network/messages/SendInfoGameForSpectatorMessage.java	271
src/main/java/lo23/battleship/online/network/messages/SendNewSpectatorMessage.java	272
src/main/java/lo23/battleship/online/network/messages/SendTextMessage.java	272
src/main/java/lo23/battleship/online/network/messages/ShotNotificationMessage.java	272
src/main/java/lo23/battleship/online/network/messages/ShotNotificationResultForSpectatorMessage.java	273
src/main/java/lo23/battleship/online/network/messages/ShotNotificationResultMessage.java	273
src/main/java/lo23/battleship/online/network/messages/UpdateGameMessage.java	273
src/main/java/structData/Boat.java	275
src/main/java/structData/BoatType.java	275
src/main/java/structData/ChatMessage.java	276
src/main/java/structData/ContactGroup.java	276
src/main/java/structData/DataUser.java	276
src/main/java/structData/Game.java	276
src/main/java/structData/MessageType.java	277
src/main/java/structData/Player.java	277
src/main/java/structData/Position.java	277
src/main/java/structData/Profile.java	277
src/main/java/structData/Shot.java	278
src/main/java/structData/StatusGame.java	278
src/main/java/structData/User.java	278

Chapter 6

Namespace Documentation

6.1 Package data

Classes

- class [CDataCom](#)
CDataCom : interface of Data for Com.
- class [CDataMain](#)
CDataCom : interface of Data for IHM Main.
- class [CDataTable](#)
Data's interface for Table.
- class [DataController](#)
Data's controller.

6.2 Package guiMain

Packages

- package [controller](#)

Classes

- class [GameCell](#)
GameCell, descendant class ListCell<Game>, display the game name and offers the possibility to join or watch a game.
- class [GuiMainController](#)
This class implements the network controller.
- interface [GuiMainInterface](#)
GUI Main Interface.
- class [IpCell](#)
IpCell, Display a list of Ips on a list that can be removed with a button.
- class [PlayerCell](#)
PlayerCell, descendant class ListCell<User>, display the user name and offers the possibility watch his profil.

6.3 Package guiMain.controller

Classes

- class [ChangeProfileController](#)
- class [CreateGameController](#)
- class [IpConfigController](#)
 - This class implements the controller of the Ip Configuration page.*
- class [LoginController](#)
- class [menuController](#)
 - This class call the lobby view and enable interaction between button and method.*
- class [ProfilController](#)
 - [GameCell](#), implement interface Initializable, display the user profil.*
- class [SignupController](#)
 - This class implements the controller of the SignUp page.*
- class [WaitingRoomController](#)

6.4 Package guiTable

Packages

- package [controllers](#)

Classes

- class [BoatDrawing](#)
 - Boat drawing : class to draw a boat.*
- class [CaseDrawing](#)
 - [CaseDrawing](#) : class to draw a case.*
- interface [GuiTableInterface](#)
 - Interface for the Ihm Table team.*

6.5 Package guiTable.controllers

Classes

- class [BaseController](#)
 - every class controller will herit from [BaseController](#)*
- class [BelgianPlacementPhaseController](#)
 - implementation of placement controller for belgian phase*
- class [ChatController](#)
- class [ClassicPlacementPhaseController](#)
 - implementation of placement controller for classic phase*
- class [gameInterface](#)
 - [gameInterface](#)*
- class [GamePhaseController](#)
 - [GamePhaseController](#).*

- class [GuiTableController](#)
GuiTableController.
- class [ObservationPhase](#)
- class [ObserverPhaseController](#)
ObserverPhaseController.
- class [PlacementPhaseController](#)
PlacementPhaseController.
- class [ProfileController](#)

6.6 Package interfacesData

Classes

- interface [IDataCom](#)
Data's interface for Com.
- interface [IDataMain](#)
Data's interface for Main.
- interface [IDataTable](#)
Data's interface for IHM-Table.

6.7 Package Io23

Packages

- package [battleship](#)

6.8 Package Io23.battleship

Packages

- package [online](#)

Classes

- class [MainApp](#)

6.9 Package Io23.battleship.online

Packages

- package [network](#)
Created by xzirva on 17/10/17.

6.10 Package `lo23.battleship.online.network`

Created by xzirva on 17/10/17.

Packages

- package [messages](#)

Created by xzirva on 17/10/17.

Classes

- interface [COMInterface](#)

The [COMInterface](#) compiles the services offered by the network package.

- class [NetworkController](#)

This class implements the network controller.

- class [NetworkListener](#)

This class implements network listeners.

- class [NetworkModuleInterface](#)

This class implements the [COMInterface](#) interface and the different methods (services) it offers.

- class [NetworkSender](#)

This class implements network senders.

- class [NetworkServer](#)

This class implements the network server.

6.10.1 Detailed Description

Created by xzirva on 17/10/17.

This package contains classes needed to exchange message through the network `NetWorkModuleInterface` Net

6.11 Package `lo23.battleship.online.network.messages`

Created by xzirva on 17/10/17.

Classes

- class [ConnectionEstablishedMessage](#)
This class implements the message which is sent when when a user disconnects from the application.
- class [ConnectionRequestMessage](#)
Network message class, extends [Message](#) class.
- class [CreatedGameNotificationMessage](#)
Network message class, extends [Message](#) class.
- class [DisconnectionMessage](#)
This class implements the message which is sent when when a user disconnects from the application.
- class [GameQuitMessage](#)
This class implements the message sent when a game is removed from the network (and the list of games) by his creator(user) or when the game is over.
- class [GameQuitSpectatorMessage](#)
This class implements the message which is sent when a spectator quit a game he is watching.
- class [GameWonMessage](#)
This class implements the message which is sent when a game is over with a victory of one of the players.
- class [GameWonMessageToSpectator](#)
This class implements the message which is sent when a game is over with a victory of one of the players.
- class [GetInfoGameForSpectatorMessage](#)
This class implements the message which is sent when when a user wants to watch a game.
- class [GetProfileRequestMessage](#)
This class implements the message which is sent when when a user wants to see the profile(including statistics) of an other user.
- class [JoinGameRequestMessage](#)
This class implements the message which is sent when when a user wants to join a game created by another user.
- class [JoinGameResponseMessage](#)
This class implements the message which is sent when when a creator of a game adds a second player to the game.
- class [Message](#)
This abstract class implements the core structure of a message sent on the network.
- class [NotifyReadyMessage](#)
This class implements the message sent by the local user (player) to the other player when he is ready to play.
- class [ProfileRequestAnswerMessage](#)
Network message class, extends [Message](#) class.
- class [SendInfoGameForSpectatorMessage](#)
This class implements the message sent when the local user has been accepted as spectator.
- class [SendNewSpectatorMessage](#)
This class implements the message sent when a user has been accepted as spectator by the creator of the game.
- class [SendTextMessage](#)
This class implements textual messages sent/received by the local user to/from a distant user.
- class [ShotNotificationMessage](#)
This class implements the message sent by the local user (player) when he shots.
- class [ShotNotificationResultForSpectatorMessage](#)
This class implements the message sent by the local user (player) to the spectators with the result of the other player shot (missed, touched or sunk boat).
- class [ShotNotificationResultMessage](#)
This class implements the message sent by the local user (player) to the other player with the result of the other player's shot (missed, touched or sunk boat).
- class [UpdateGameMessage](#)
This class implements the message sent by the local user to other users when a game (namely the status of the game) has changed.

6.11.1 Detailed Description

Created by xzirva on 17/10/17.

This package contains all types of message

6.12 Package structData

Classes

- class [Boat](#)
Boat is the class which represents a ship of a player.
- enum [BoatType](#)
BoatType is an enumerate of all the types of boats and numbur of cases associated.
- class [ChatMessage](#)
ChatMessage is a class for chat messages.
- class [ContactGroup](#)
ContactGroup is a class a contact groups.
- class [DataUser](#)
DataUser class : inherits from User.
- class [Game](#)
Game is a class for the user's game.
- enum [MessageType](#)
Enumerate for the types of messages (Class Message)
- class [Player](#)
Player is a class for players in a game.
- class [Position](#)
Position is a class of a position in a table.
- class [Profile](#)
Profile is a class for the user's profile.
- class [Shot](#)
Shot is a class of a shot of a player.
- enum [StatusGame](#)
StatusGame is an enumerate of all the status of games.
- class [User](#)
Class User.

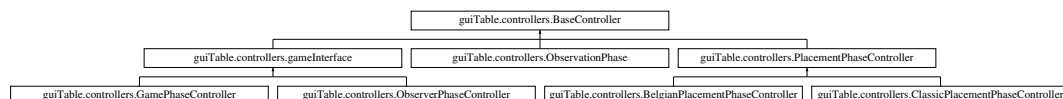
Chapter 7

Class Documentation

7.1 guiTable.controllers.BaseController Class Reference

every class controller will herit from [BaseController](#)

Inheritance diagram for guiTable.controllers.BaseController:



Public Member Functions

- [ChatController](#) [fillChatSlot](#) (AnchorPane pane, String chatFXMLUrl, String conversation)
[fillChatSlot\(\)](#) allows external class to fill the chatPane and get the [ChatController](#)

Static Public Attributes

- static final int [GRID_X](#) = 100
- static final int [GRID_Y](#) = 100
- static final int [SPACE](#) = 3
- static final int [GRID_ELEMENT_SIZE](#) = 35
- static final int [NB_CASES_GRID](#) = 10

Protected Member Functions

- FXMLLoader [fillElement](#) (AnchorPane paneToFill, String contentAdress)
Allows to replace pane by another one.

7.1.1 Detailed Description

every class controller will herit from [BaseController](#)

7.1.2 Member Function Documentation

- ##### 7.1.2.1 ChatController guiTable.controllers.BaseController.fillChatSlot (AnchorPane pane, String chatFXMLUrl, String conversation)

[fillChatSlot\(\)](#) allows external class to fill the chatPane and get the [ChatController](#)

Parameters

<i>pane</i>	
<i>chatFXMLUrl</i>	
<i>conversation</i>	

Returns

chatController

7.1.2.2 FXMLLoader guiTable.controllers.BaseController.fillElement (AnchorPane *paneToFill*, String *contentAdress*)
[protected]

Allows to replace pane by another one.

Parameters

<i>paneToFill</i>	
<i>contentAdress</i>	

Returns

FXMLLoader

7.1.3 Member Data Documentation

7.1.3.1 final int guiTable.controllers.BaseController.GRID_ELEMENT_SIZE = 35 [static]

7.1.3.2 final int guiTable.controllers.BaseController.GRID_X = 100 [static]

7.1.3.3 final int guiTable.controllers.BaseController.GRID_Y = 100 [static]

7.1.3.4 final int guiTable.controllers.BaseController.NB_CASES_GRID = 10 [static]

7.1.3.5 final int guiTable.controllers.BaseController.SPACE = 3 [static]

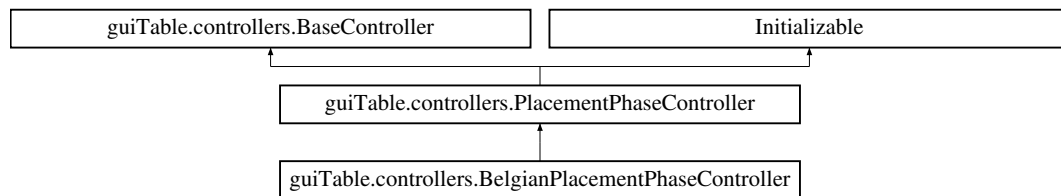
The documentation for this class was generated from the following file:

- src/main/java/guiTable/controllers/[BaseController.java](#)

7.2 guiTable.controllers.BelgianPlacementPhaseController Class Reference

implementation of placement controller for belgian phase

Inheritance diagram for guiTable.controllers.BelgianPlacementPhaseController:



Protected Member Functions

- void [initBoatMap](#) ()

Protected Attributes

- Rectangle [cuirasseRectangle](#)

Private Attributes

- Rectangle [croiseurRectangle1](#)
- Rectangle [croiseurRectangle2](#)
- Rectangle [torpilleurRectangle1](#)
- Rectangle [torpilleurRectangle2](#)
- Rectangle [torpilleurRectangle3](#)
- Rectangle [sousMarinRectangle1](#)
- Rectangle [sousMarinRectangle2](#)
- Rectangle [sousMarinRectangle3](#)
- Rectangle [sousMarinRectangle4](#)

Additional Inherited Members

7.2.1 Detailed Description

implementation of placement controller for belgian phase

7.2.2 Member Function Documentation

7.2.2.1 void `guiTable.controllers.BelgianPlacementPhaseController.initBoatMap ()` [protected]

7.2.3 Member Data Documentation

7.2.3.1 Rectangle `guiTable.controllers.BelgianPlacementPhaseController.croiseurRectangle1` [private]

7.2.3.2 Rectangle `guiTable.controllers.BelgianPlacementPhaseController.croiseurRectangle2` [private]

7.2.3.3 Rectangle `guiTable.controllers.BelgianPlacementPhaseController.cuirasseRectangle` [protected]

7.2.3.4 Rectangle `guiTable.controllers.BelgianPlacementPhaseController.sousMarinRectangle1` [private]

7.2.3.5 Rectangle `guiTable.controllers.BelgianPlacementPhaseController.sousMarinRectangle2` [private]

7.2.3.6 Rectangle `guiTable.controllers.BelgianPlacementPhaseController.sousMarinRectangle3` [private]

7.2.3.7 Rectangle `guiTable.controllers.BelgianPlacementPhaseController.sousMarinRectangle4` [private]

7.2.3.8 Rectangle `guiTable.controllers.BelgianPlacementPhaseController.torpilleurRectangle1` [private]

7.2.3.9 Rectangle `guiTable.controllers.BelgianPlacementPhaseController.torpilleurRectangle2` [private]

7.2.3.10 Rectangle `guiTable.controllers.BelgianPlacementPhaseController.torpilleurRectangle3` [private]

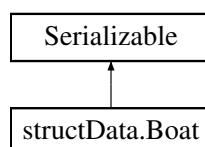
The documentation for this class was generated from the following file:

- [src/main/java/guiTable/controllers/BelgianPlacementPhaseController.java](#)

7.3 structData.Boat Class Reference

[Boat](#) is the class which represents a ship of a player.

Inheritance diagram for structData.Boat:



Public Member Functions

- [Boat](#) ()
Constructor by default status = false means that the boat isn't sunk.
- [Boat](#) ([BoatType](#) typedata, Boolean rotation, [Position](#) pos)
Constructor with parameters.
- [Boat](#) ([BoatType](#) typedata, Boolean statusdata, List< [Position](#) > listCasesdata)
Constructor with parameters.
- [BoatType](#) [getType](#) ()
Accessor for the type.
- Boolean [getSunk](#) ()
Accessor for the status.
- List< [Position](#) > [getListCases](#) ()
Accessor for the List of cases.
- void [setType](#) ([BoatType](#) typedata)
Mutator for Type.
- void [setSunk](#) (Boolean statusdata)
Mutator for the boat's status.
- void [setListcases](#) (List< [Position](#) > listCasesdata)
Mutator for the boat's list of positions.
- void [setListcases](#) (Boolean rotation, [Position](#) pos)
Mutator for listcases : the list of cases.
- [Boat](#) [cloneBoat](#) ([Boat](#) boatclone)
Clone method for a boat.
- void [addPosition](#) ([Position](#) pos)
Add a position to listCases.
- [Boat](#) [updateShot](#) ([Position](#) shot)
Verify if a position belongs to a boat.
- Boolean [verifyBoatStatus](#) ()
verify is the boat has been sunk

Static Package Attributes

- static final long [serialVersionUID](#) = 1L

Private Attributes

- [BoatType](#) type
- Boolean isSunk
- List< [Position](#) > listCases

7.3.1 Detailed Description

[Boat](#) is the class which represents a ship of a player.

7.3.2 Constructor & Destructor Documentation

7.3.2.1 structData.Boat.Boat ()

Constructor by default status = false means that the boat isn't sunk.

7.3.2.2 structData.Boat.Boat ([BoatType](#) typedata, Boolean rotation, [Position](#) pos)

Constructor with parameters.

Parameters

<i>typedata</i>	: the type of the new boat
<i>rotation</i>	: is the boat is rotated
<i>pos</i>	: the boat's position

7.3.2.3 structData.Boat.Boat (BoatType *typedata*, Boolean *statusdata*, List< Position > *listCasesdata*)

Constructor with parameters.

Parameters

<i>typedata</i>	: the type of the new boat
<i>statusdata</i>	: the status of the new boat
<i>listCasesdata</i>	: the list of positions of the new boat

7.3.3 Member Function Documentation

7.3.3.1 void structData.Boat.addPosition (Position *pos*)

Add a position to listCases.

Parameters

<i>pos</i>	position to add
------------	-----------------

7.3.3.2 Boat structData.Boat.cloneBoat (Boat *boatclone*)

Clone method for a boat.

Parameters

<i>boatclone</i>	: boat to clone
------------------	-----------------

Returns

a clone of the boat

7.3.3.3 List<Position> structData.Boat.getListCases ()

Accessor for the List of cases.

Returns

the boat's list of cases

7.3.3.4 Boolean structData.Boat.getSunk ()

Accessor for the status.

Returns

the boat's status

7.3.3.5 BoatType structData.Boat.getType ()

Accessor for the type.

Returns

the boat's type

7.3.3.6 void structData.Boat.setListcases (List< Position > listCasesdata)

Mutator for the boat's list of positions.

Parameters

<i>listCasesdata</i>	: the new list of posiitons
----------------------	-----------------------------

7.3.3.7 void structData.Boat.setListcases (Boolean rotation, Position pos)

Mutator for listcases : the list of cases.

Parameters

<i>rotation</i>	if the boat is rotated or not
<i>pos</i>	the boat's position

7.3.3.8 void structData.Boat.setSunk (Boolean statusdata)

Mutator for the boat's status.

Parameters

<i>statusdata</i>	: the new boat's status
-------------------	-------------------------

7.3.3.9 void structData.Boat.setType (BoatType typedata)

Mutator for Type.

Parameters

<i>typedata</i>	: the new value of the type of the boat
-----------------	---

7.3.3.10 Boat structData.Boat.updateShot (Position *shot*)

Verify if a position belongs to a boat.

Parameters

<i>shot</i>	: position to check
-------------	---------------------

Returns

a boolean set to true if the position belongs to a boat

7.3.3.11 Boolean structData.Boat.verifyBoatStatus ()

verify is the boat has been sunk

Returns

the new status of the current boat

7.3.4 Member Data Documentation

7.3.4.1 Boolean structData.Boat.isSunk [private]

7.3.4.2 List<Position> structData.Boat.listCases [private]

7.3.4.3 final long structData.Boat.serialVersionUID = 1L [static],[package]

7.3.4.4 BoatType structData.Boat.type [private]

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

- src/main/java/structData/[Boat.java](#)

7.4 guiTable.BoatDrawing Class Reference

Boat drawing : class to draw a boat.

Public Member Functions

- [BoatDrawing](#) ([BoatType](#) boatType, [Rectangle](#) boatRectangle)
Constructor of a object.
- boolean [isActive](#) ()
- void [setActive](#) (boolean active)
- boolean [isRotation](#) ()
- void [setRotation](#) (boolean rotation)
- [Rectangle](#) [getBoatRectangle](#) ()
- void [setBoatRectangle](#) ([Rectangle](#) boatRectangle)
- double [getInitialLayoutX](#) ()
- double [getInitialLayoutY](#) ()
- Integer [getGridRow](#) ()
- Integer [getGridCol](#) ()
- [BoatType](#) [getBoatType](#) ()
- void [setBoatType](#) ([BoatType](#) boatType)
- boolean [isPlaced](#) ()
- [Paint](#) [getActiveColor](#) ()
- [Paint](#) [getDisactiveColor](#) ()
- int [getBoatSize](#) ()
- [Paint](#) [getBadPlacementColor](#) ()
- void [reinit](#) ()
Reiniciates the boat, meaning that it has returned to its original position.
- [BoatDrawing](#) [setActiveBoat](#) ([HashMap](#)< [Rectangle](#), [BoatDrawing](#) > boatMap)
Activates the boat.
- void [setPosition](#) (Integer colIndex, Integer rowIndex)
- void [setPlaced](#) (Boolean bool)

Private Attributes

- boolean active
- boolean rotation
- [Rectangle](#) boatRectangle
- [BoatType](#) boatType
- Integer gridRow
- Integer gridCol
- Boolean placed
- final double initialLayoutX
- final double initialLayoutY
- final [Color](#) activeColor = [Color.web](#)("#d8d875")
- final [Color](#) disactiveColor = [Color.web](#)("#ababab")
- final [Color](#) badPlacementColor = [Color.CRIMSON](#)

7.4.1 Detailed Description

Boat drawing : class to draw a boat.

7.4.2 Constructor & Destructor Documentation

7.4.2.1 guiTable.BoatDrawing.BoatDrawing ([BoatType](#) boatType, [Rectangle](#) boatRectangle)

Constructor of a object.

Parameters

<i>boatType</i>	The type of the boat.
<i>boatRectangle</i>	The object Rectangle associated with this boat.

7.4.3 Member Function Documentation

7.4.3.1 Paint guiTable.BoatDrawing.getActiveColor ()

Returns

the activeColor

7.4.3.2 Paint guiTable.BoatDrawing.getBadPlacementColor ()

7.4.3.3 Rectangle guiTable.BoatDrawing.getBoatRectangle ()

7.4.3.4 int guiTable.BoatDrawing.getBoatSize ()

7.4.3.5 BoatType guiTable.BoatDrawing.getBoatType ()

7.4.3.6 Paint guiTable.BoatDrawing.getDisactiveColor ()

Returns

the disactiveColor

7.4.3.7 Integer guiTable.BoatDrawing.getGridCol ()

7.4.3.8 Integer guiTable.BoatDrawing.getGridRow ()

7.4.3.9 double guiTable.BoatDrawing.getInitialLayoutX ()

7.4.3.10 double guiTable.BoatDrawing.getInitialLayoutY ()

7.4.3.11 boolean guiTable.BoatDrawing.isActive ()

7.4.3.12 boolean guiTable.BoatDrawing.isPlaced ()

7.4.3.13 boolean guiTable.BoatDrawing.isRotation ()

7.4.3.14 void guiTable.BoatDrawing.reinit ()

Reiniciates the boat, meaning that it has returned to its original position.

7.4.3.15 void guiTable.BoatDrawing.setActive (boolean *active*)7.4.3.16 BoatDrawing guiTable.BoatDrawing.setActiveBoat (HashMap< Rectangle, BoatDrawing > *boatMap*)

Activates the boat.

Parameters

<i>boatMap</i>	HashMap containing all the boats.
----------------	-----------------------------------

Returns

This (the active boat).

7.4.3.17 void guiTable.BoatDrawing.setBoatRectangle (Rectangle *boatRectangle*)

7.4.3.18 void guiTable.BoatDrawing.setBoatType (BoatType *boatType*)

7.4.3.19 void guiTable.BoatDrawing.setPlaced (Boolean *bool*)

7.4.3.20 void guiTable.BoatDrawing.setPosition (Integer *colIndex*, Integer *rowIndex*)

7.4.3.21 void guiTable.BoatDrawing.setRotation (boolean *rotation*)

7.4.4 Member Data Documentation

7.4.4.1 boolean guiTable.BoatDrawing.active [private]

7.4.4.2 final Color guiTable.BoatDrawing.activeColor = Color.web("#d8d875") [private]

7.4.4.3 final Color guiTable.BoatDrawing.badPlacementColor = Color.CRIMSON [private]

7.4.4.4 Rectangle guiTable.BoatDrawing.boatRectangle [private]

7.4.4.5 BoatType guiTable.BoatDrawing.boatType [private]

7.4.4.6 final Color guiTable.BoatDrawing.disactiveColor = Color.web("#ababab") [private]

7.4.4.7 Integer guiTable.BoatDrawing.gridCol [private]

7.4.4.8 Integer guiTable.BoatDrawing.gridRow [private]

7.4.4.9 final double guiTable.BoatDrawing.initialLayoutX [private]

7.4.4.10 final double guiTable.BoatDrawing.initialLayoutY [private]

7.4.4.11 Boolean guiTable.BoatDrawing.placed [private]

7.4.4.12 boolean guiTable.BoatDrawing.rotation [private]

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

- src/main/java/guiTable/[BoatDrawing.java](#)

7.5 structData.BoatType Enum Reference

[BoatType](#) is an enumerate of all the types of boats and numbur of cases associated.

Public Member Functions

- String [getName](#) ()
Accessor for the boat's name.
- int [getNbCases](#) ()
Accessor for the boat's nbCases.

Public Attributes

- [PORTEAVIONS](#) =("porte-avions", 5)
- [CROISEURFR](#) =("croiseur", 4)
- [CONTRETORPILLEUR](#) =("contre-torpilleur", 3)
- [SOUSMARINFR](#) =("sous-marin", 3)
- [TORPILLEUR](#) =("torpilleur", 2)
- [CUIRASSE](#) =("cuirassé", 4)
- [CROISEURB](#) =("croiseur", 3)
- [SOUSMARINB](#) =("sous-marin", 1)

Private Member Functions

- [BoatType](#) (String [name](#), int [nbCases](#))

Private Attributes

- final String [name](#)
- final int [nbCases](#)

7.5.1 Detailed Description

[BoatType](#) is an enumerate of all the types of boats and numbur of cases associated.

7.5.2 Constructor & Destructor Documentation

7.5.2.1 `structData.BoatType.BoatType (String name, int nbCases)` `[private]`

7.5.3 Member Function Documentation

7.5.3.1 `String structData.BoatType.getName ()`

Accessor for the boat's name.

Returns

the name

7.5.3.2 `int structData.BoatType.getNbCases ()`

Accessor for the boat's nbCases.

Returns

the nbCases : the number of the boat's number of cases

7.5.4 Member Data Documentation

7.5.4.1 `structData.BoatType.CONTRETORPILLEUR =("contre-torpilleur", 3)`

7.5.4.2 `structData.BoatType.CROISEURB =("croiseur", 3)`

7.5.4.3 `structData.BoatType.CROISEURFR =("croiseur", 4)`

7.5.4.4 `structData.BoatType.CUIRASSE =("cuirassé", 4)`

7.5.4.5 `final String structData.BoatType.name` [private]

7.5.4.6 `final int structData.BoatType.nbCases` [private]

7.5.4.7 `structData.BoatType.PORTEAVIONS =("porte-avions", 5)`

7.5.4.8 `structData.BoatType.SOUSMARINB =("sous-marin", 1)`

7.5.4.9 `structData.BoatType.SOUSMARINFR =("sous-marin", 3)`

7.5.4.10 `structData.BoatType.TORPILLEUR =("torpilleur", 2)`

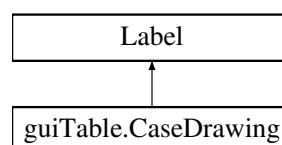
The documentation for this enum was generated from the following file:

- `src/main/java/structData/BoatType.java`

7.6 guiTable.CaseDrawing Class Reference

[CaseDrawing](#) : class to draw a case.

Inheritance diagram for guiTable.CaseDrawing:



Classes

- enum [Type](#)

Public Member Functions

- [CaseDrawing](#) ()
- [CaseDrawing](#) ([Type](#) t)

7.6.1 Detailed Description

[CaseDrawing](#) : class to draw a case.

7.6.2 Constructor & Destructor Documentation

7.6.2.1 `guiTable.CaseDrawing.CaseDrawing ()`

7.6.2.2 `guiTable.CaseDrawing.CaseDrawing (Type t)`

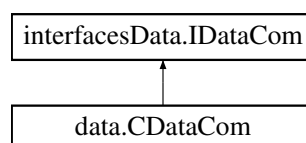
The documentation for this class was generated from the following file:

- `src/main/java/guiTable/CaseDrawing.java`

7.7 data.CDataCom Class Reference

[CDataCom](#) : interface of Data for Com.

Inheritance diagram for data.CDataCom:



Public Member Functions

- [CDataCom](#) ([DataController](#) dc)
- void [setInterfaceMain](#) ([GuiMainInterface](#) i)
- void [setInterfaceTable](#) ([GuiTableInterface](#) i)
- void [setInterfaceCom](#) ([COMInterface](#) c)
- [Game](#) [getCreatedGame](#) ()
Returns the current Game.
- void [setGameJoinResponse](#) (Boolean ok, [Player](#) player1, [Player](#) player2)
The distant user has accepted or not the request to join the game and the method updateGameData will be used to update the game data.
- void [setGameJoinResponse](#) (Boolean no)
The distance user has refused the request to join the game.
- void [addUserToUserList](#) ([User](#) u)
After an user has connected, this user will be added to the list of user.
- void [sendStatistics](#) ([Profile](#) profile)
Sends the profile of a distant user to the local user.
- void [notifToJoinGame](#) ([Profile](#) sender, [Game](#) g)
Add the player to the game if it is available.
- void [addNewGameList](#) ([Game](#) g)
Adds the game given as a parameter to the list of games.
- void [removeGameFromList](#) ([Game](#) g)
Removes the game given as a parameter from the list of games.
- void [receiveMessage](#) ([ChatMessage](#) message)
Takes the chat message given as a parameter in order to transmit it to IHM-Table.
- void [receiveReady](#) ()
Indicates that a player is ready to play (all his boats are placed on his table) so that the shots phase can be displayed.
- void [coordinates](#) ([Shot](#) s)
Takes a Shot to transmit it to IHM-Table.
- void [coordinates](#) ([Shot](#) s, [Boat](#) b)
Takes a Shot and in option a Boat to transmit it to IHM-Table.
- void [notifyAttendedGameWon](#) ([Player](#) p)
Notify when a player won a game.
- void [updateAttendedGame](#) ([Player](#) p, [Shot](#) s, [Boat](#) b)
Update attended in a game.
- [Profile](#) [getUserProfile](#) ()
Returns the local user's profile.
- void [changeStatusGame](#) ([Game](#) g)
Takes a game given as a parameter and updates its status.
- [User](#) [getLocalUser](#) ()
Accessor for the local User.
- void [setLocalGame](#) ([Game](#) g)
Set the local Game with the game given as a parameter.
- void [removeUser](#) ([User](#) u)
To remove a User.
- void [removeGame](#) ([Game](#) g)
Remove a Game from local list.
- void [notifiedGameWon](#) ()
Notification that you won, update stats and display win.
- void [notifyToSpecGame](#) ([User](#) spec)
Notify that a new spectator has joined the game.

- void [newRequestSpectator](#) ([User](#) u)
A new spectator want to join the game, he need to get the informations of the game.
- void [joinGameSpectator](#) ([Game](#) g)
For a User to join a game as a spectator.
- void [notifyQuitSpectator](#) ([User](#) spec)
Notifies everybody when a spectator quitted the game.
- [Player](#) [getOtherPlayer](#) ()
Accessor for the game's other player.
- int [getPlayerPosition](#) ([Player](#) p, [Game](#) g)
Accessor for a player's position.

Private Attributes

- final [DataController](#) controller
- [GuiMainInterface](#) interfaceMain
- [GuiTableInterface](#) interfaceTable
- [COMInterface](#) interfaceCom

7.7.1 Detailed Description

[CDataCom](#) : interface of Data for Com.

7.7.2 Constructor & Destructor Documentation

7.7.2.1 [data.CDataCom.CDataCom](#) ([DataController](#) dc)

7.7.3 Member Function Documentation

7.7.3.1 void [data.CDataCom.addNewGameList](#) ([Game](#) g)

Adds the game given as a parameter to the list of games.

Parameters

<i>g</i>	: The new game
----------	----------------

Implements [interfacesData.IDataCom](#).

7.7.3.2 void [data.CDataCom.addUserToUserList](#) ([User](#) u)

After an user has connected, this user will be added to the list of user.

Parameters

<i>u</i>	: The new user
----------	----------------

Implements [interfacesData.IDataCom](#).

7.7.3.3 void data.CDataCom.changeStatusGame (Game *g*)

Takes a game given as a parameter and updates its status.

Parameters

<i>g</i>	: the game which status has been modified
----------	---

Implements [interfacesData.IDataCom](#).

7.7.3.4 void data.CDataCom.coordinates (Shot *s*)

Takes a Shot to transmit it to IHM-Table.

Parameters

<i>s</i>	: The position played by the user
----------	-----------------------------------

Implements [interfacesData.IDataCom](#).

7.7.3.5 void data.CDataCom.coordinates (Shot *s*, Boat *b*)

Takes a Shot and in option a Boat to transmit it to IHM-Table.

Parameters

<i>s</i>	: The position played by the user
<i>b</i>	: In option, the boat that was sunk

Implements [interfacesData.IDataCom](#).

7.7.3.6 Game data.CDataCom.getCreatedGame ()

Returns the current Game.

Returns

the current Game

Implements [interfacesData.IDataCom](#).

7.7.3.7 User data.CDataCom.getLocalUser ()

Accessor for the local User.

Returns

public void [removeUser](#)(User u)

Implements [interfacesData.IDataCom](#).

7.7.3.8 Player data.CDataCom.getOtherPlayer ()

Accessor for the game's other player.

Returns

the other player

Implements [interfacesData.IDataCom](#).

7.7.3.9 int data.CDataCom.getPlayerPosition (Player p, Game g)

Accessor for a player's position.

Parameters

<i>p</i>	player
<i>g</i>	game

Returns

the player's position

Implements [interfacesData.IDataCom](#).

7.7.3.10 Profile data.CDataCom.getUserProfile ()

Returns the local user's profile.

Returns

the local user's profile

Implements [interfacesData.IDataCom](#).

7.7.3.11 void data.CDataCom.joinGameSpectator (Game g)

For a User to join a game as a spectator.

Parameters

<i>g</i>	the game
----------	----------

Implements [interfacesData.IDataCom](#).

7.7.3.12 void data.CDataCom.newRequestSpectator (User *u*)

A new spectator want to join the game, he need to get the informations of the game.

Parameters

<i>u</i>	The spectator who want to come
----------	--------------------------------

Implements [interfacesData.IDataCom](#).

7.7.3.13 void data.CDataCom.notifiedGameWon ()

Notification that you won, update stats and display win.

Implements [interfacesData.IDataCom](#).

7.7.3.14 void data.CDataCom.notifToJoinGame (Profile *sender*, Game *g*)

Add the player to the game if it is available.

Parameters

<i>sender</i>	: The player who sends this request
<i>g</i>	: The game that the player wants to join

Implements [interfacesData.IDataCom](#).

7.7.3.15 void data.CDataCom.notifyAttendedGameWon (Player *p*)

Notify when a player won a game.

Parameters

<i>p</i>	the player
----------	------------

Implements [interfacesData.IDataCom](#).

7.7.3.16 void data.CDataCom.notifyQuitSpectator (User *spec*)

Notifies everybody when a spectator quitted the game.

Parameters

<i>spec</i>	the spectator that quitted
-------------	----------------------------

Implements [interfacesData.IDataCom](#).

7.7.3.17 void data.CDataCom.notifyToSpecGame (User *spec*)

Notify that a new spectator has joined the game.

Parameters

<i>spec</i>	New spectator
-------------	---------------

Implements [interfacesData.IDataCom](#).

7.7.3.18 void data.CDataCom.receiveMessage (ChatMessage *message*)

Takes the chat message given as a parameter in order to transmit it to IHM-Table.

Parameters

<i>message</i>	: The chat message to transmit
----------------	--------------------------------

Implements [interfacesData.IDataCom](#).

7.7.3.19 void data.CDataCom.receiveReady ()

Indicates that a player is ready to play (all his boats are placed on his table) so that the shots phase can be displayed.

Implements [interfacesData.IDataCom](#).

7.7.3.20 void data.CDataCom.removeGame (Game *g*)

Remove a Game from local list.

Parameters

<i>g</i>	: Game to remove
----------	------------------

Implements [interfacesData.IDataCom](#).

7.7.3.21 void data.CDataCom.removeGameFromList (Game *g*)

Removes the game given as a parameter from the list of games.

Parameters

<i>g</i>	: game to remove
----------	------------------

Implements [interfacesData.IDataCom](#).

7.7.3.22 void data.CDataCom.removeUser (User *u*)

To remove a User.

Parameters

<i>u</i>	User to remove
----------	----------------

Implements [interfacesData.IDataCom](#).

7.7.3.23 void data.CDataCom.sendStatistics (Profile *profile*)

Sends the profile of a distant user to the local user.

Parameters

<i>profile</i>	: the profile of distant user
----------------	-------------------------------

Implements [interfacesData.IDataCom](#).

7.7.3.24 void data.CDataCom.setGameJoinResponse (Boolean *ok*, Player *player1*, Player *player2*)

The distant user has accepted or not the request to join the game and the method `updateGameData` will be used to update the game data.

Parameters

<i>ok</i>	: Acceptance of the request to join the game
<i>player1</i>	: Creator of the game
<i>player2</i>	: The player who joins the game

Implements [interfacesData.IDataCom](#).

7.7.3.25 void data.CDataCom.setGameJoinResponse (Boolean *no*)

The distance user has refused the request to join the game.

Parameters

<i>no</i>	: Refuse of the request to join the game
-----------	--

Implements [interfacesData.IDataCom](#).

7.7.3.26 void data.CDataCom.setInterfaceCom (COMInterface *c*)

7.7.3.27 void data.CDataCom.setInterfaceMain (GuiMainInterface *i*)

7.7.3.28 void data.CDataCom.setInterfaceTable (GuiTableInterface *i*)

7.7.3.29 void data.CDataCom.setLocalGame (Game *g*)

Set the local Game with the game given as a parameter.

Parameters

<i>g</i>	: new value for the local Game
----------	--------------------------------

Implements [interfacesData.IDataCom](#).

7.7.3.30 void data.CDataCom.updateAttendedGame (Player *p*, Shot *s*, Boat *b*)

Update attended in a game.

Parameters

<i>p</i>	a player
<i>s</i>	a shot
<i>b</i>	a boat

Implements [interfacesData.IDataCom](#).

7.7.4 Member Data Documentation

7.7.4.1 final DataController data.CDataCom.controller [private]

7.7.4.2 COMInterface data.CDataCom.interfaceCom [private]

7.7.4.3 GuiMainInterface data.CDataCom.interfaceMain [private]

7.7.4.4 GuiTableInterface data.CDataCom.interfaceTable [private]

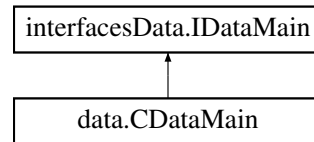
The documentation for this class was generated from the following file:

- src/main/java/data/[CDataCom.java](#)

7.8 data.CDataMain Class Reference

[CDataCom](#) : interface of Data for IHM Main.

Inheritance diagram for data.CDataMain:



Public Member Functions

- [CDataMain](#) ([DataController](#) dc)
- void [setInterfaceCom](#) ([COMInterface](#) i)
- void [editProfile](#) (String username, String password, String avatar, String lastName, String firstName, Date birthDate)
 - Modify the local profile.*
- void [createAccount](#) (String login, String username, HashSet ips, int port, String password, List< [ContactGroup](#) > contactList, String avatar, String lastname, String firstname, Date birthDate)
 - Create a local account.*
- [Profile](#) [getLocalProfile](#) ()
- void [notifGameChosen](#) ([Game](#) g)
 - Notifies the away application that an user wants to join the game given as parameter.*
- void [askDisconnection](#) ()
 - Notifies away applications that the local user disconnects and erases his session.*
- Boolean [connection](#) (String login, String password) throws [UnknownHostException](#)
 - Loads the saved data of the user and researches players.*
- [Game](#) [newGame](#) (Boolean newClassicType, String newName, Boolean newHumanOpponent, Integer newTimePerShot, Integer newTimeToPlaceBoats, Boolean newSpectator, Boolean newSpectatorChat)
 - Add a new game to the list of games.*
- void [removeGame](#) ([Game](#) g)
 - Remove a Game from local list.*
- List< [Game](#) > [getGames](#) ()
- void [setLocalGame](#) ([Game](#) g)
 - Set the local game.*
- void [getProfile](#) ([User](#) u)
 - Returns the profile of an user.*
- void [setListIps](#) (HashSet ips)
 - Set the list of IP adress.*
- void [gameToSpec](#) ([Game](#) g)
 - Add a spectator in the game.*
- void [setPort](#) (int p)
 - Set the port.*
- List< [User](#) > [getListUsers](#) ()
 - Get the list of connected users.*
- void [clear](#) ()

Private Attributes

- [DataController](#) controller
- [COMInterface](#) interfaceCom

7.8.1 Detailed Description

[CDataCom](#) : interface of Data for IHM Main.

7.8.2 Constructor & Destructor Documentation

7.8.2.1 `data.CDataMain.CDataMain (DataController dc)`

7.8.3 Member Function Documentation

7.8.3.1 `void data.CDataMain.askDisconnection ()`

Notifies away applications that the local user disconnects and erases his session.

Implements [interfacesData.IDataMain](#).

7.8.3.2 `void data.CDataMain.clear ()`

Implements [interfacesData.IDataMain](#).

7.8.3.3 `Boolean data.CDataMain.connection (String login, String password)` throws `UnknownHostException`

Loads the saved data of the user and researches players.

Parameters

<i>login</i>	the login
<i>password</i>	the password

Returns

a boolean

Exceptions

<code><i>java.net.UnknownHostException</i></code>	
---	--

Implements [interfacesData.IDataMain](#).

7.8.3.4 void data.CDataMain.createAccount (String *login*, String *username*, HashSet *ips*, int *port*, String *password*, List<ContactGroup> *contactList*, String *avatar*, String *lastname*, String *firstname*, Date *birthDate*)

Create a local account.

Parameters

<i>login</i>	: the user's login
<i>username</i>	: the user's username
<i>ips</i>	: list of the IP addresses known by the user
<i>port</i>	: the port
<i>password</i>	: the user's password
<i>contactList</i>	: list of the user's contacts
<i>avatar</i>	: a path to the user's avatar
<i>lastname</i>	: the user's lastname
<i>firstname</i>	: the user's firstname
<i>birthDate</i>	: the user's birthdate

Implements [interfacesData.IDataMain](#).

7.8.3.5 void data.CDataMain.editProfile (String *username*, String *password*, String *avatar*, String *lastName*, String *firstName*, Date *birthDate*)

Modify the local profile.

Parameters

<i>username</i>	: new username
<i>password</i>	: new password
<i>avatar</i>	: new avatar
<i>lastName</i>	: new lastName
<i>firstName</i>	: new firstName
<i>birthDate</i>	: new birthDate

Implements [interfacesData.IDataMain](#).

7.8.3.6 void data.CDataMain.gameToSpec (Game *g*)

Add a spectator in the game.

Parameters

<i>g</i>	: game that the spectator wants to join
----------	---

Implements [interfacesData.IDataMain](#).

7.8.3.7 List<Game> data.CDataMain.getGames ()

Returns

the local list of games

Implements [interfacesData.IDataMain](#).

7.8.3.8 List<User> data.CDataMain.getListUsers ()

Get the list of connected users.

Returns

the list of connected users

Implements [interfacesData.IDataMain](#).

7.8.3.9 Profile data.CDataMain.getLocalProfile ()**Returns**

the local profile

Implements [interfacesData.IDataMain](#).

7.8.3.10 void data.CDataMain.getProfile (User u)

Returns the profile of an user.

Parameters

<i>u</i>	the user
----------	----------

Implements [interfacesData.IDataMain](#).

7.8.3.11 Game data.CDataMain.newGame (Boolean *newClassicType*, String *newName*, Boolean *newHumanOpponent*, Integer *newTimePerShot*, Integer *newTimeToPlaceBoats*, Boolean *newSpectator*, Boolean *newSpectatorChat*)

Add a new game to the list of games.

Parameters

<i>newClassicType</i>	the type
<i>newName</i>	the name
<i>newHumanOpponent</i>	if is against a human or not
<i>newTimePerShot</i>	the time per shot
<i>newTimeToPlaceBoats</i>	the time to place boats
<i>newSpectator</i>	if spectators are allowed
<i>newSpectatorChat</i>	if chat is allowed

Returns

the new game

Implements [interfacesData.IDataMain](#).

7.8.3.12 void data.CDataMain.notifGameChosen (Game *g*)

Notifies the away application that an user wants to join the game given as parameter.

Parameters

<i>g</i>	: the game the user wants to join
----------	-----------------------------------

Implements [interfacesData.IDataMain](#).

7.8.3.13 void data.CDataMain.removeGame (Game *g*)

Remove a Game from local list.

Parameters

<i>g</i>	: Game to remove
----------	------------------

Implements [interfacesData.IDataMain](#).

7.8.3.14 void data.CDataMain.setInterfaceCom (COMInterface *i*)**7.8.3.15 void data.CDataMain.setListIps (HashSet *lps*)**

Set the list of IP adress.

Parameters

<i>lps</i>	: list of IP
------------	--------------

Implements [interfacesData.IDataMain](#).

7.8.3.16 void data.CDataMain.setLocalGame (Game *g*)

Set the local game.

Parameters

<i>g</i>	: game to set
----------	---------------

Implements [interfacesData.IDataMain](#).

7.8.3.17 void `data.CDataMain.setPort (int p)`

Set the port.

Parameters

<code>p</code>	: the port to set
----------------	-------------------

Implements [interfacesData.IDataMain](#).

7.8.4 Member Data Documentation

7.8.4.1 **DataController** `data.CDataMain.controller` [private]

7.8.4.2 **COMInterface** `data.CDataMain.interfaceCom` [private]

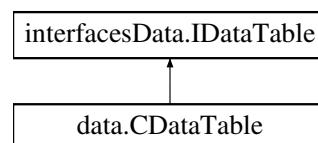
The documentation for this class was generated from the following file:

- `src/main/java/data/CDataMain.java`

7.9 data.CDataTable Class Reference

Data's interface for Table.

Inheritance diagram for `data.CDataTable`:



Public Member Functions

- `CDataTable (DataController dc)`
- void `setInterfaceTable (GuiTableInterface t)`
Set the table's interface.
- void `setInterfaceCom (COMInterface c)`
Set the com's interface.
- void `setInterfaceMain (GuiMainInterface m)`
Set the main's interface.
- Boolean `exit ()`
Function to exit.

- void [textMessage](#) (String message)
Add the message to the chat in the current game.
- void [coordinate](#) ([Position](#) pos)
Point out the position of shot.
- void [coordinateShips](#) (List< [Boat](#) > listBoat)
Point out the the boats that players place at the beginning of game.
- [Game](#) [getLocalGame](#) ()
- void [changeStatusGameStarted](#) ()
Transmit to every user that the game is beign played, allow them to spectate.
- void [gameEnded](#) ()
To end a game.
- void [timerOver](#) ()
- [Game](#) [getObserverGame](#) ()
Accessor for the observed game.

Static Protected Attributes

- static final int [NB_CASES_GRID](#) = 10

Private Attributes

- final [DataController](#) controller
- [GuiTableInterface](#) interfaceTable
- [GuiMainInterface](#) interfaceMain
- [COMInterface](#) interfaceCom

7.9.1 Detailed Description

Data's interface for Table.

7.9.2 Constructor & Destructor Documentation

7.9.2.1 data.CDataTable.CDataTable ([DataController](#) dc)

7.9.3 Member Function Documentation

7.9.3.1 void data.CDataTable.changeStatusGameStarted ()

Transmit to every user that the game is beign played, allow them to spectate.

Implements [interfacesData.IDataTable](#).

7.9.3.2 void data.CDataTable.coordinate ([Position](#) pos)

Point out the position of shot.

Parameters

<i>pos</i>	: The position of shot.
------------	-------------------------

Implements [interfacesData.IDataTable](#).

7.9.3.3 void data.CDataTable.coordinateShips (List< Boat > *listBoat*)

Point out the the boats that players place at the beginning of game.

Parameters

<i>listBoat</i>	: List of boats.
-----------------	------------------

Implements [interfacesData.IDataTable](#).

7.9.3.4 Boolean data.CDataTable.exit ()

Function to exit.

Returns

: 1 if the game was successfully closed and return 0 if not.

Implements [interfacesData.IDataTable](#).

7.9.3.5 void data.CDataTable.gameEnded ()

To end a game.

Implements [interfacesData.IDataTable](#).

7.9.3.6 Game data.CDataTable.getLocalGame ()

Returns

the local game

Implements [interfacesData.IDataTable](#).

7.9.3.7 Game data.CDataTable.getObserverGame ()

Accessor for the observed game.

Returns

the game observed

Implements [interfacesData.IDataTable](#).

7.9.3.8 void data.CDataTable.setInterfaceCom (COMInterface *c*)

Set the com's interface.

Parameters

<i>c</i>	interface com
----------	---------------

7.9.3.9 void data.CDataTable.setInterfaceMain (*GuiMainInterface m*)

Set the main's interface.

Parameters

<i>m</i>	interface main
----------	----------------

7.9.3.10 void data.CDataTable.setInterfaceTable (*GuiTableInterface t*)

Set the table's interface.

Parameters

<i>t</i>	Table interface
----------	-----------------

7.9.3.11 void data.CDataTable.textMessage (*String message*)

Add the message to the chat in the current game.

Parameters

<i>message</i>	: The main part of message that the player wants to send.
----------------	---

Implements [interfacesData.IDDataTable](#).

7.9.3.12 void data.CDataTable.timerOver ()

Implements [interfacesData.IDDataTable](#).

7.9.4 Member Data Documentation

7.9.4.1 final DataController data.CDataTable.controller [private]

7.9.4.2 COMInterface data.CDataTable.interfaceCom [private]

7.9.4.3 GuiMainInterface data.CDataTable.interfaceMain [private]

7.9.4.4 **GuiTableInterface** `data.CDataTable.interfaceTable` `[private]`

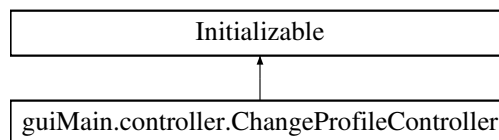
7.9.4.5 **final int** `data.CDataTable.NB_CASES_GRID = 10` `[static], [protected]`

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

- `src/main/java/data/CDataTable.java`

7.10 guiMain.controller.ChangeProfileController Class Reference

Inheritance diagram for `guiMain.controller.ChangeProfileController`:



Public Member Functions

- void `initialize` (URL location, ResourceBundle resources)
- void `init` (`User` user)
Initialize profile data of the correspondent local user.
- void `setMainController` (`GuiMainController` c)
Set `GuiMainController` mainController.

Package Functions

- void `backToMenu` (`ActionEvent` event)
Return to menu window.
- void `modifyAvatar` (`ActionEvent` event)
Open dialog to select avatar image.
- void `saveProfile` (`ActionEvent` event)
Save data changed into user profile.

Private Member Functions

- void `openFile` (`File` file)
Open image file and set avatar.

Private Attributes

- [GuiMainController](#) `mainController`
- Label `nameTitle`
- TextField `userName`
- TextField `firstName`
- TextField `lastName`
- PasswordField `userPassword`
- DatePicker `birthdate`
- Label `numberOfGame`
- Label `numberOfGameWon`
- Label `numberOfGameLost`
- Button `modifyAvatarButton`
- Button `backButton`
- Button `saveButton`
- ImageView `userAvatar`
- Label `errorMessage`
- String `avatarPath`

7.10.1 Member Function Documentation

7.10.1.1 void guiMain.controller.ChangeProfileController.backToMenu (*ActionEvent event*) [package]

Return to menu window.

Parameters

<i>event</i>	: backButton event
--------------	------------------------------------

7.10.1.2 void guiMain.controller.ChangeProfileController.init (*User user*)

Initialize profile data of the correspondent local user.

Parameters

<i>user</i>	
-------------	--

7.10.1.3 void guiMain.controller.ChangeProfileController.initialize (*URL location*, *ResourceBundle resources*)

7.10.1.4 void guiMain.controller.ChangeProfileController.modifyAvatar (*ActionEvent event*) [package]

Open dialog to select avatar image.

Parameters

<i>event</i>	: modifyAvatarButton event
--------------	--

7.10.1.5 void guiMain.controller.ChangeProfileController.openFile (File *file*) [private]

Open image file and set avatar.

Parameters

<i>file</i>	: image file selected by the user
-------------	-----------------------------------

7.10.1.6 void guiMain.controller.ChangeProfileController.saveProfile ([ActionEvent](#) *event*) [package]

Save data changed into user profile.

Parameters

<i>event</i>	: saveButton event
--------------	------------------------------------

7.10.1.7 void guiMain.controller.ChangeProfileController.setMainController ([GuiMainController](#) *c*)

Set [GuiMainController](#) mainController.

Parameters

<i>c</i>	: GuiMainController
----------	-------------------------------------

7.10.2 Member Data Documentation

7.10.2.1 String guiMain.controller.ChangeProfileController.avatarPath [private]

7.10.2.2 Button guiMain.controller.ChangeProfileController.backButton [private]

7.10.2.3 DatePicker guiMain.controller.ChangeProfileController.birthdate [private]

7.10.2.4 Label guiMain.controller.ChangeProfileController.errorMessage [private]

7.10.2.5 TextField guiMain.controller.ChangeProfileController.firstName [private]

7.10.2.6 TextField guiMain.controller.ChangeProfileController.lastName [private]

7.10.2.7 [GuiMainController](#) guiMain.controller.ChangeProfileController.mainController [private]

7.10.2.8 Button guiMain.controller.ChangeProfileController.modifyAvatarButton [private]

7.10.2.9 Label guiMain.controller.ChangeProfileController.nameTitle [private]

- 7.10.2.10 Label `guiMain.controller.ChangeProfileController.numberOfGame` [private]
- 7.10.2.11 Label `guiMain.controller.ChangeProfileController.numberOfGameLost` [private]
- 7.10.2.12 Label `guiMain.controller.ChangeProfileController.numberOfGameWon` [private]
- 7.10.2.13 Button `guiMain.controller.ChangeProfileController.saveButton` [private]
- 7.10.2.14 ImageView `guiMain.controller.ChangeProfileController.userAvatar` [private]
- 7.10.2.15 TextField `guiMain.controller.ChangeProfileController.userName` [private]
- 7.10.2.16 PasswordField `guiMain.controller.ChangeProfileController.userPassword` [private]

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

- `src/main/java/guiMain/controller/ChangeProfileController.java`

7.11 guiTable.controllers.ChatController Class Reference

Public Member Functions

- void `init` (String conv)
- void `createConversation` (ArrayList< [ChatMessage](#) > list)
créer la conversation à partir de la liste des messages
- void `sendMyMessage` ()
- void `receiveAMessage` ([ChatMessage](#) message)
- void `reloadConversation` ()
- void `setDataController` ([IDataTable](#) d)
- String `getConversation` ()
- void `doProfileArea` ()

Private Attributes

- TextArea `conversationArea`
- TextField `field`
- Label `profils`
- [IDataTable](#) `dataController`
- String `conversation`

7.11.1 Member Function Documentation

- 7.11.1.1 void `guiTable.controllers.ChatController.createConversation` (ArrayList< [ChatMessage](#) > list)

créer la conversation à partir de la liste des messages

Parameters

<i>list</i>	
-------------	--

7.11.1.2 void guiTable.controllers.ChatController.doProfileArea ()

7.11.1.3 String guiTable.controllers.ChatController.getConversation ()

7.11.1.4 void guiTable.controllers.ChatController.init (String *conv*)

7.11.1.5 void guiTable.controllers.ChatController.receiveAMessage (ChatMessage *message*)

7.11.1.6 void guiTable.controllers.ChatController.reloadConversation ()

7.11.1.7 void guiTable.controllers.ChatController.sendMyMessage ()

7.11.1.8 void guiTable.controllers.ChatController.setDataController (IDataTable *d*)

7.11.2 Member Data Documentation

7.11.2.1 String guiTable.controllers.ChatController.conversation [private]

7.11.2.2 TextArea guiTable.controllers.ChatController.conversationArea [private]

7.11.2.3 IDataTable guiTable.controllers.ChatController.dataController [private]

7.11.2.4 TextField guiTable.controllers.ChatController.field [private]

7.11.2.5 Label guiTable.controllers.ChatController.profiles [private]

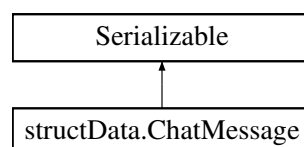
The documentation for this class was generated from the following file:

- [src/main/java/guiTable/controllers/ChatController.java](#)

7.12 structData.ChatMessage Class Reference

[ChatMessage](#) is a class for chat messages.

Inheritance diagram for structData.ChatMessage:



Public Member Functions

- [ChatMessage](#) ([User](#) newSender, String newMessage, Date newBirthdate)
Constructor with all parameters.
- [User](#) [getProfile](#) ()
Accessors for profile(user)
- String [getContent](#) ()
Accessors for content.
- Date [getTime](#) ()
Accessors for time.
- void [setProfile](#) ([User](#) u)
Mutator for profile(user)
- void [setContent](#) (String m)
Mutator for content.
- void [setTime](#) (Date t)
Mutator for time.
- [ChatMessage](#) [cloneChatMessage](#) ([ChatMessage](#) chatMessageClone)
Clone function to copy a chatMessage.

Static Package Attributes

- static final long [serialVersionUID](#) = 2L

Private Attributes

- [User](#) [profile](#)
- String [content](#)
- Date [time](#)

7.12.1 Detailed Description

[ChatMessage](#) is a class for chat messages.

7.12.2 Constructor & Destructor Documentation

7.12.2.1 structData.ChatMessage.ChatMessage ([User](#) newSender, String newMessage, Date newBirthdate)

Constructor with all parameters.

Parameters

<i>newSender</i>	: person who send this message
<i>newMessage</i>	:the message has to be send
<i>newBirthdate</i>	: the birthdate of this message

7.12.3 Member Function Documentation

7.12.3.1 ChatMessage structData.ChatMessage.cloneChatMessage (ChatMessage chatMessageClone)

Clone function to copy a chatMessage.

Parameters

<i>chatMessageClone</i>	: message to be cloned
-------------------------	------------------------

Returns

a copied chatMessage

7.12.3.2 String structData.ChatMessage.getContent ()

Accessors for content.

Returns

the content of message

7.12.3.3 User structData.ChatMessage.getProfile ()

Accessors for profile(user)

Returns

the user who send the message

7.12.3.4 Date structData.ChatMessage.getTime ()

Accessors for time.

Returns

the time that the message has been sent

7.12.3.5 void structData.ChatMessage.setContent (String m)

Mutator for content.

Parameters

<i>m</i>	: message
----------	-----------

7.12.3.6 void structData.ChatMessage.setProfile (User *u*)

Mutator for profile(user)

Parameters

<i>u</i>	: the sender
----------	--------------

7.12.3.7 void structData.ChatMessage.setTime (Date *t*)

Mutator for time.

Parameters

<i>t</i>	: time that message has been sent
----------	-----------------------------------

7.12.4 Member Data Documentation

7.12.4.1 String structData.ChatMessage.content [private]

7.12.4.2 User structData.ChatMessage.profile [private]

7.12.4.3 final long structData.ChatMessage.serialVersionUID = 2L [static], [package]

7.12.4.4 Date structData.ChatMessage.time [private]

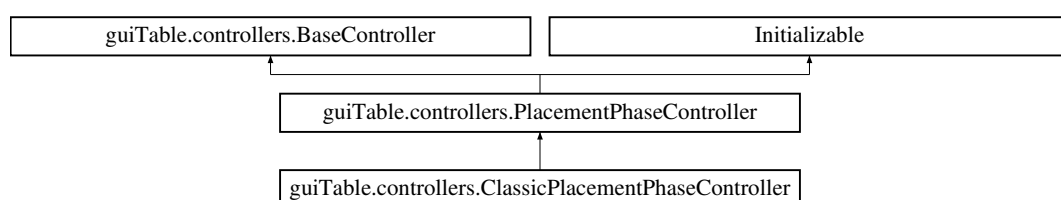
The documentation for this class was generated from the following file:

- src/main/java/structData/[ChatMessage.java](#)

7.13 guiTable.controllers.ClassicPlacementPhaseController Class Reference

implementation of placement controller for classic phase

Inheritance diagram for guiTable.controllers.ClassicPlacementPhaseController:



Protected Member Functions

- void [initBoatMap](#) ()

Private Attributes

- Rectangle [porteAvionsRectangle](#)
- Rectangle [croiseurRectangle](#)
- Rectangle [contreTorpilleurRectangle](#)
- Rectangle [sousMarinRectangle](#)
- Rectangle [torpilleurRectangle](#)

Additional Inherited Members

7.13.1 Detailed Description

implementation of placement controller for classic phase

7.13.2 Member Function Documentation

7.13.2.1 void `guiTable.controllers.ClassicPlacementPhaseController.initBoatMap ()` [protected]

7.13.3 Member Data Documentation

7.13.3.1 Rectangle `guiTable.controllers.ClassicPlacementPhaseController.contreTorpilleurRectangle` [private]

7.13.3.2 Rectangle `guiTable.controllers.ClassicPlacementPhaseController.croiseurRectangle` [private]

7.13.3.3 Rectangle `guiTable.controllers.ClassicPlacementPhaseController.porteAvionsRectangle` [private]

7.13.3.4 Rectangle `guiTable.controllers.ClassicPlacementPhaseController.sousMarinRectangle` [private]

7.13.3.5 Rectangle `guiTable.controllers.ClassicPlacementPhaseController.torpilleurRectangle` [private]

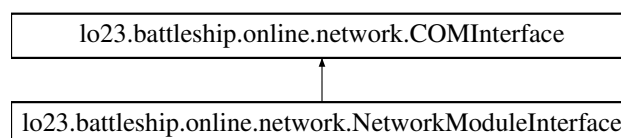
The documentation for this class was generated from the following file:

- `src/main/java/guiTable/controllers/ClassicPlacementPhaseController.java`

7.14 Io23.battleship.online.network.COMInterface Interface Reference

The [COMInterface](#) compiles the services offered by the network package.

Inheritance diagram for `Io23.battleship.online.network.COMInterface`:



Public Member Functions

- void [notifyReady](#) ([User](#) user, [Player](#) playerToNotify)
Notifies if a player is ready or not.
- void [sendChatMessage](#) ([ChatMessage](#) chatMessage, [Game](#) g)
Sends a chat message.
- void [getProfile](#) ([User](#) user)
Retrieves an user profile.
- void [changeStatusGame](#) ([Game](#) game)
Update game object (containing game.status)
- void [notifyNewGame](#) ([Game](#) g)
notify a new game
- void [joinGame](#) ([Game](#) g)
allow an user to join a game
- void [notifyJoinGameResponse](#) (boolean isOk, [Profile](#) user, [Game](#) g)
allow an user to join a game
- void [askDisconnection](#) ()
allow an user to be disconnected to the network
- void [sendShot](#) ([Player](#) player, [Game](#) g, [Shot](#) shot)
send a shot from a player on the right game
- void [coordinates](#) ([Player](#) destPlayer, [Shot](#) resultShot, [Game](#) game, [Boat](#) boat)
send a shot result to a player on the right game
- void [searchForPlayers](#) ()
Initiates network discovery.
- void [removeGame](#) ([Game](#) game)
notifies every user to remove the game from their list
- void [notifyGameWon](#) ()
notifies every user to remove the game from their list
- void [getInfoGameForSpectator](#) ([Player](#) player, [User](#) spec)
spectator send a request to a player in order to have the game informations
- void [sendInfoGameForSpectator](#) ([Game](#) game, [User](#) spec)
player send the game to the spectator after his request
- void [sendNewSpectator](#) ([User](#) u, [Player](#) p, [HashSet](#)< [User](#) > listSpectator)
alert everybody (the other player and the spectator) that there is a new spectator
- void [gameQuitSpectator](#) ([User](#) spec, [Game](#) game)
alert everybody that the spectator quit the game
- void [clearNetwork](#) ()
Clears network state : removes all players from local user list of connected players.

7.14.1 Detailed Description

The [COMInterface](#) compiles the services offered by the network package.

It should be implemented by the actual class which will be use to call those services

7.14.2 Member Function Documentation

7.14.2.1 void Io23.battleship.online.network.COMInterface.askDisconnection ()

allow an user to be disconnected to the network

Implemented in [Io23.battleship.online.network.NetworkModuleInterface](#).

7.14.2.2 void `lo23.battleship.online.network.COMInterface.changeStatusGame (Game game)`

Update game object (containing game.status)

Parameters

<i>game</i>	: Game of which the status changed
-------------	------------------------------------

Implemented in [lo23.battleship.online.network.NetworkModuleInterface](#).

7.14.2.3 void `lo23.battleship.online.network.COMInterface.clearNetwork ()`

Clears network state : removes all players from local user list of connected players.

Implemented in [lo23.battleship.online.network.NetworkModuleInterface](#).

7.14.2.4 void `lo23.battleship.online.network.COMInterface.coordinates (Player destPlayer, Shot resultShot, Game game, Boat boat)`

send a shot result to a player on the right game

Parameters

<i>destPlayer</i>	destination player
<i>game</i>	where the ships are
<i>resultShot</i>	result of the shot
<i>boat</i>	optional

Implemented in [lo23.battleship.online.network.NetworkModuleInterface](#).

7.14.2.5 void `lo23.battleship.online.network.COMInterface.gameQuitSpectator (User spec, Game game)`

alert everybody that the spectator quit the game

Parameters

<i>spec</i>	spectator who quit the game
<i>game</i>	game quit by the spectator (permit to retrieve player and listSpectator)

Implemented in [lo23.battleship.online.network.NetworkModuleInterface](#).

7.14.2.6 void `lo23.battleship.online.network.COMInterface.getInfoGameForSpectator (Player player, User spec)`

spectator send a request to a player in order to have the game informations

Parameters

<i>player</i>	player game
<i>spec</i>	spectator who ask for request

Implemented in [lo23.battleship.online.network.NetworkModuleInterface](#).

7.14.2.7 void lo23.battleship.online.network.COMInterface.getProfile (User user)

Retrieves an user profile.

Parameters

<i>user</i>	: User of which local user asks for profile
-------------	---

Implemented in [lo23.battleship.online.network.NetworkModuleInterface](#).

7.14.2.8 void lo23.battleship.online.network.COMInterface.joinGame (Game g)

allow an user to join a game

Parameters

<i>g</i>	Game the local user wants to join
----------	-----------------------------------

Implemented in [lo23.battleship.online.network.NetworkModuleInterface](#).

7.14.2.9 void lo23.battleship.online.network.COMInterface.notifyGameWon ()

notifies every user to remove the game from their list

Implemented in [lo23.battleship.online.network.NetworkModuleInterface](#).

7.14.2.10 void lo23.battleship.online.network.COMInterface.notifyJoinGameResponse (boolean isOk, Profile user, Game g)

allow an user to join a game

Parameters

<i>isOk</i>	access to game true=access granted false= access denied
<i>user</i>	who asked to join the game
<i>g</i>	Game <i>user</i> joined if isOk

Implemented in [lo23.battleship.online.network.NetworkModuleInterface](#).

7.14.2.11 void `lo23.battleship.online.network.COMInterface.notifyNewGame (Game g)`

notify a new game

Parameters

<i>g</i>	: newly created game with one player
----------	--------------------------------------

Implemented in [lo23.battleship.online.network.NetworkModuleInterface](#).

7.14.2.12 void `lo23.battleship.online.network.COMInterface.notifyReady (User user, Player playerToNotify)`

Notifies if a player is ready or not.

Parameters

<i>user</i>	player who is notify
-------------	----------------------

Implemented in [lo23.battleship.online.network.NetworkModuleInterface](#).

7.14.2.13 void `lo23.battleship.online.network.COMInterface.removeGame (Game game)`

notifies every user to remove the game from their list

Parameters

<i>game</i>	Game to delete
-------------	----------------

Implemented in [lo23.battleship.online.network.NetworkModuleInterface](#).

7.14.2.14 void `lo23.battleship.online.network.COMInterface.searchForPlayers ()`

Initiates network discovery.

Search for players who are connected

Implemented in [lo23.battleship.online.network.NetworkModuleInterface](#).

7.14.2.15 void `lo23.battleship.online.network.COMInterface.sendChatMessage (ChatMessage chatMessage, Game g)`

Sends a chat message.

Parameters

<i>chatMessage</i>	message send
<i>g</i>	: game related to the message

Implemented in [lo23.battleship.online.network.NetworkModuleInterface](#).

7.14.2.16 void lo23.battleship.online.network.COMInterface.sendInfoGameForSpectator (Game *game*, User *spec*)

player send the game to the spectator after his request

Parameters

<i>game</i>	game send to the spectator
<i>spec</i>	spectator who want to see the game

Implemented in [lo23.battleship.online.network.NetworkModuleInterface](#).

7.14.2.17 void lo23.battleship.online.network.COMInterface.sendNewSpectator (User *u*, Player *p*, HashSet< User > *listSpectator*)

alert everybody (the other player and the spectator) that there is a new spectator

Parameters

<i>u</i>	new spectator
<i>p</i>	other player
<i>listSpectator</i>	list of all actual spectators

Implemented in [lo23.battleship.online.network.NetworkModuleInterface](#).

7.14.2.18 void lo23.battleship.online.network.COMInterface.sendShot (Player *player*, Game *g*, Shot *shot*)

send a shot from a player on the right game

Parameters

<i>player</i>	:	Player who send the shot
<i>g</i>	:	Game where the ships are
<i>shot</i>	:	Shot where the player shot

Implemented in [lo23.battleship.online.network.NetworkModuleInterface](#).

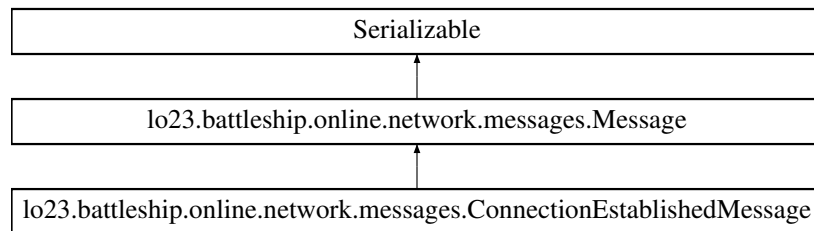
The documentation for this interface was generated from the following file:

- [src/main/java/lo23/battleship/online/network/COMInterface.java](#)

7.15 lo23.battleship.online.network.messages.ConnectionEstablishedMessage Class Reference

This class implements the message which is sent when when a user disconnects from the application.

Inheritance diagram for lo23.battleship.online.network.messages.ConnectionEstablishedMessage:



Public Member Functions

- String [getType](#) ()
Message type getter.
- void [process](#) (IDataCom IData, InetAddress senderAddress)
Updates the network state using the [NetworkController](#) and the list of connected users using the IDataCom interface.

Package Functions

- [ConnectionEstablishedMessage](#) (User sender, List< InetAddress > ipTable, Game game)
Class constructor.

Private Attributes

- User sender
- List< InetAddress > [ipAdressesTable](#)
- Game [createdGame](#)

Additional Inherited Members

7.15.1 Detailed Description

This class implements the message which is sent when when a user disconnects from the application.

The user who wants to disconnect sends this message to all connected users.

This class extends the abstract [Message](#) and implements the two abstract methods: `getType` and `process`

See also

[Message](#)

Author

COM Module

7.15.2 Constructor & Destructor Documentation

7.15.2.1 Io23.battleship.online.network.messages.ConnectionEstablishedMessage.ConnectionEstablishedMessage (User sender, List< InetAddress > ipTable, Game game) [package]

Class constructor.

Parameters

<i>sender</i>	is the User class of the network message sender.
<i>ipTable</i>	is the IP addresses list known by the sender.
<i>game</i>	is the game the sender has created, if he has created a game. Thus, the receiver can fill its displayed game list once connected to the P2P network.

7.15.3 Member Function Documentation

7.15.3.1 String Io23.battleship.online.network.messages.ConnectionEstablishedMessage.getType ()

[Message](#) type getter.

Implementation of an abstract method.

Returns

type, this is the message type.

7.15.3.2 void Io23.battleship.online.network.messages.ConnectionEstablishedMessage.process (IDataCom IData, InetAddress senderAddress)

Updates the network state using the [NetworkController](#) and the list of connected users using the IDataCom interface.

Then sends a [ConnectionRequestMessage](#) to the IP Addresses that the local user does not know and a

[ConnectionEstablishedMessage](#)

to the users the sender of the

[ConnectionEstablishedMessage](#)

does not know

Parameters

<i>IData</i>	interface with Data.
<i>senderAddress</i>	sender IP address.

7.15.4 Member Data Documentation

7.15.4.1 **Game** `lo23.battleship.online.network.messages.ConnectionEstablishedMessage.createdGame` `[private]`

7.15.4.2 **List<InetAddress>** `lo23.battleship.online.network.messages.ConnectionEstablishedMessage.ipAddressesTable` `[private]`

7.15.4.3 **User** `lo23.battleship.online.network.messages.ConnectionEstablishedMessage.sender` `[private]`

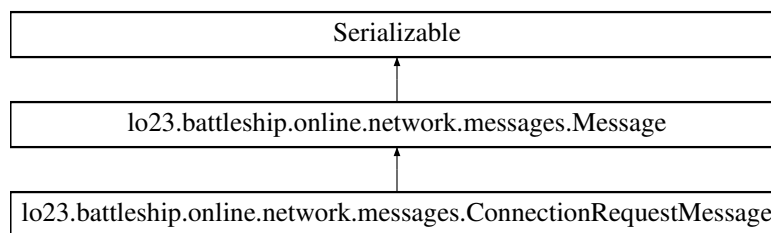
The documentation for this class was generated from the following file:

- `src/main/java/lo23/battleship/online/network/messages/ConnectionEstablishedMessage.java`

7.16 `lo23.battleship.online.network.messages.ConnectionRequestMessage` Class Reference

Network message class, extends `Message` class.

Inheritance diagram for `lo23.battleship.online.network.messages.ConnectionRequestMessage`:



Public Member Functions

- `ConnectionRequestMessage (User user, List< InetAddress > ipAddressesTable, Game game)`
Class constructor.
- `String getType ()`
Returns the type of the message.
- `void process (IDataCom IData, InetAddress senderAddress)`
Method updating the `NetworkController` and class members accordingly to the message sent.

Private Attributes

- `User sender`
- `List< InetAddress > ipAddressesTable`
Known IP addresses list.
- `Game game`

Additional Inherited Members

7.16.1 Detailed Description

Network message class, extends [Message](#) class.

[Message](#) sent when the local user wants to join a network of P2P users on the application. The local user (the sender) let the other users know he wants to join the network.

7.16.2 Constructor & Destructor Documentation

7.16.2.1 Io23.battleship.online.network.messages.ConnectionRequestMessage.ConnectionRequestMessage ([User](#) *user*, [List](#)< [InetAddress](#) > *ipAdressesTable*, [Game](#) *game*)

Class constructor.

Parameters

<i>user</i>	is the User class of the network message sender.
<i>ipAdressesTable</i>	is the IP addresses list known by the sender.

7.16.3 Member Function Documentation

7.16.3.1 [String](#) Io23.battleship.online.network.messages.ConnectionRequestMessage.getType ()

Returns the type of the message.

Implementation of an abstract method.

Returns

type :

[String](#)

7.16.3.2 void Io23.battleship.online.network.messages.ConnectionRequestMessage.process ([IDataCom](#) *IData*, [InetAddress](#) *senderAddress*)

Method updating the [NetworkController](#) and class members accordingly to the message sent.

Parameters

<i>IData</i>	interface with Data.
<i>senderAddress</i>	sender IP address.

7.16.4 Member Data Documentation

7.16.4.1 **Game** `lo23.battleship.online.network.messages.ConnectionRequestMessage.game` `[private]`

7.16.4.2 **List<InetAddress>** `lo23.battleship.online.network.messages.ConnectionRequestMessage.ipAddressesTable`
`[private]`

Known IP addresses list.

7.16.4.3 **User** `lo23.battleship.online.network.messages.ConnectionRequestMessage.sender` `[private]`

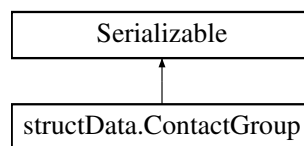
The documentation for this class was generated from the following file:

- `src/main/java/lo23/battleship/online/network/messages/ConnectionRequestMessage.java`

7.17 structData.ContactGroup Class Reference

`ContactGroup` is a class a contact groups.

Inheritance diagram for `structData.ContactGroup`:



Public Member Functions

- `ContactGroup ()`
Constructor by default.

Static Package Attributes

- static final long `serialVersionUID` = 3L

Private Attributes

- List< `User` > `listUsers`
- int `rights`
- String `name`

7.17.1 Detailed Description

`ContactGroup` is a class a contact groups.

7.17.2 Constructor & Destructor Documentation

7.17.2.1 structData.ContactGroup.ContactGroup ()

Constructor by default.

7.17.3 Member Data Documentation

7.17.3.1 List<User> structData.ContactGroup.listUsers [private]

7.17.3.2 String structData.ContactGroup.name [private]

7.17.3.3 int structData.ContactGroup.rights [private]

7.17.3.4 final long structData.ContactGroup.serialVersionUID = 3L [static], [package]

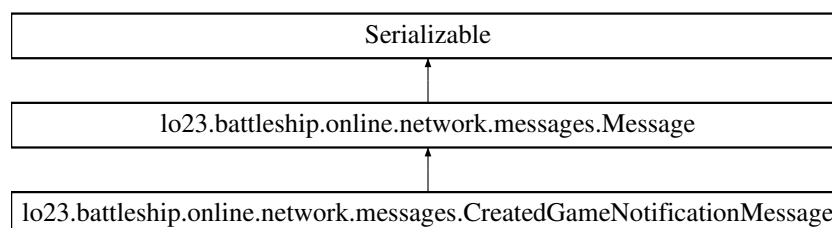
The documentation for this class was generated from the following file:

- [src/main/java/structData/ContactGroup.java](#)

7.18 lo23.battleship.online.network.messages.CreatedGameNotificationMessage Class Reference

Network message class, extends [Message](#) class.

Inheritance diagram for lo23.battleship.online.network.messages.CreatedGameNotificationMessage:



Public Member Functions

- [CreatedGameNotificationMessage](#) ([Game](#) gameCreated)
Class constructor.
- String [getType](#) ()
Returns the type of the message.
- void [process](#) ([IDataCom](#) IData, InetAddress senderAddress)
Unused method for this class.

Package Attributes

- [Game game](#)

7.18.1 Detailed Description

Network message class, extends [Message](#) class.

[Message](#) notifying Data a new game has been created by the sender.

7.18.2 Constructor & Destructor Documentation

7.18.2.1 `lo23.battleship.online.network.messages.CreatedGameNotificationMessage.CreatedGameNotificationMessage (Game gameCreated)`

Class constructor.

Parameters

<i>gameCreated</i>	game created by the sender. The receiver will be able to fill its displayed game list.
--------------------	--

7.18.3 Member Function Documentation

7.18.3.1 `String lo23.battleship.online.network.messages.CreatedGameNotificationMessage.getType ()`

Returns the type of the message.

Implementation of an abstract method.

Returns

type :

`String`

7.18.3.2 `void lo23.battleship.online.network.messages.CreatedGameNotificationMessage.process (IDataCom IData, InetAddress senderAddress)`

Unused method for this class.

Parameters

<i>IData</i>	interface with Data.
<i>senderAddress</i>	sender IP address.

7.18.4 Member Data Documentation

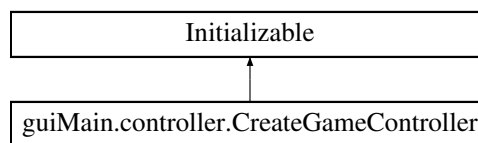
7.18.4.1 Game `lo23.battleship.online.network.messages.CreatedGameNotificationMessage.game` [package]

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

- `src/main/java/lo23/battleship/online/network/messages/`[CreatedGameNotificationMessage.java](#)

7.19 guiMain.controller.CreateGameController Class Reference

Inheritance diagram for `guiMain.controller.CreateGameController`:



Public Member Functions

- void [initialize](#) (URL url, ResourceBundle rb)
Initializes the controller class.
- void [setMainController](#) ([GuiMainController](#) c)
Set main controller.

Package Functions

- void [newGame](#) (ActionEvent event) throws IOException
Get form data and calls IDataMain to create new game.
- void [returnToMenu](#) (ActionEvent event)
Change window to menu window.
- void [timerDisable](#) (ActionEvent event)
Disable time choice.

Private Attributes

- TextField [gameName](#)
- RadioButton [classicGameType](#)
- ToggleGroup [gameType](#)
- RadioButton [belgiumGameType](#)
- CheckBox [spectatorsAutorise](#)
- CheckBox [chatAutorise](#)
- CheckBox [timeDisabled](#)
- TextField [positioningTime](#)
- TextField [reflectionTime](#)
- Label [errorMessage](#)
- Button [createGame](#)
- Button [returnButton](#)
- [GuiMainController](#) [mainController](#)

7.19.1 Member Function Documentation

7.19.1.1 void guiMain.controller.CreateGameController.initialize (URL *url*, ResourceBundle *rb*)

Initializes the controller class.

Parameters

<i>url</i>	
<i>rb</i>	

7.19.1.2 void guiMain.controller.CreateGameController.newGame (*ActionEvent event*) throws *IOException* [package]

Get form data and calls IDataMain to create new game.

Parameters

<i>event</i>	: button #creategame event click
--------------	----------------------------------

Exceptions

<i>IOException</i>	
--------------------	--

7.19.1.3 void guiMain.controller.CreateGameController.returnToMenu (*ActionEvent event*) [package]

Change window to menu window.

Parameters

<i>event</i>	: returnButton event click
--------------	--

7.19.1.4 void guiMain.controller.CreateGameController.setMainController (*GuiMainController c*)

Set main controller.

Parameters

<i>c</i>	
----------	--

7.19.1.5 void guiMain.controller.CreateGameController.timerDisable (*ActionEvent event*) [package]

Disable time choice.

Parameters

<i>event</i>	: timeDisabled checkbox select
--------------	--

7.19.2 Member Data Documentation

7.19.2.1 `RadioButton guiMain.controller.CreateGameController.belgiumGameType` `[private]`

7.19.2.2 `CheckBox guiMain.controller.CreateGameController.chatAutorise` `[private]`

7.19.2.3 `RadioButton guiMain.controller.CreateGameController.classicGameType` `[private]`

7.19.2.4 `Button guiMain.controller.CreateGameController.createGame` `[private]`

7.19.2.5 `Label guiMain.controller.CreateGameController.errorMessage` `[private]`

7.19.2.6 `TextField guiMain.controller.CreateGameController.gameName` `[private]`

7.19.2.7 `ToggleGroup guiMain.controller.CreateGameController.gameType` `[private]`

7.19.2.8 `GuiMainController guiMain.controller.CreateGameController.mainController` `[private]`

7.19.2.9 `TextField guiMain.controller.CreateGameController.positioningTime` `[private]`

7.19.2.10 `TextField guiMain.controller.CreateGameController.reflectionTime` `[private]`

7.19.2.11 `Button guiMain.controller.CreateGameController.returnButton` `[private]`

7.19.2.12 `CheckBox guiMain.controller.CreateGameController.spectatorsAutorise` `[private]`

7.19.2.13 `CheckBox guiMain.controller.CreateGameController.timeDisabled` `[private]`

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

- `src/main/java/guiMain/controller/`[CreateGameController.java](#)

7.20 data.DataController Class Reference

Data's controller.

Public Member Functions

- [DataController](#) ()
DataController.
- void [setInterfaceMain](#) ([GuiMainInterface](#) i)
Mutator for interfaceMain.
- void [setInterfaceTable](#) ([GuiTableInterface](#) i)
Set the table's interface.
- [GuiTableInterface](#) [getTableInterface](#) ()
Accessor for table's interface.
- void [setInterfaceCom](#) ([COMInterface](#) i)
Mutator for interfaceCom.
- [CDataCom](#) [getInterfaceDataCom](#) ()
Accessor for interfaceDataCom.
- [CDataMain](#) [getInterfaceDataMain](#) ()
Accessor for interfaceDataMain.
- [CDataTable](#) [getInterfaceDataTable](#) ()
Accessor for interfaceDataTable.
- [User](#) [getLocalUser](#) ()
Accessor for the local user.
- [DataUser](#) [getLocalDataUser](#) ()
Accessor for local DataUser.
- [Profile](#) [getLocalProfile](#) ()
Accessor for local Profile.
- void [setLocalUser](#) ([User](#) u)
Mutator for User.
- void [setLocalDataUser](#) ([DataUser](#) du)
Mutator for local DataUser.
- void [setLocalProfile](#) ([Profile](#) p)
Mutator for local Profile.
- void [setLocalPlayer](#) ([Player](#) p)
Mutator for local player.
- [Player](#) [getLocalPlayer](#) ()
Accessor for local player.
- void [addUserToList](#) ([User](#) u)
Add User to local list.
- void [removeUserFromList](#) ([User](#) u)
Remove a user form local list.
- [Game](#) [getLocalGame](#) ()
Accessor local Game.
- [Game](#) [getAttendedGame](#) ()
Accessor local Game.
- void [setAttendedGame](#) ([Game](#) g)
Mutator for the attended game.
- void [setLocalGame](#) ([Game](#) g)
Mutator local Game.
- void [addGameToList](#) ([Game](#) g)
Add a Game to the local list.
- void [updateGameStatus](#) ([Game](#) g)
Update the status of game and add it into the list of Game.
- void [removeGameFromList](#) ([Game](#) g)

- Remove the game.*

 - void [updateGameData](#) (Boolean ok, [Player](#) player1, [Player](#) player2)

Used by the method setGameJoinResponse of [CDataCom](#).
- boolean [isPlayer1](#) ()

Enable to know if the local player is the player 1.
- void [updateGameDataPlaying](#) ([Shot](#) s, [Boat](#) b, boolean forLocalPlayer)

To update the game's data when players are playing.
- List< [Game](#) > [getListGames](#) ()

Get list of Games.
- List< [User](#) > [getListUsers](#) ()

Get list of Users.
- void [reloadSavedProfile](#) (String login, String mdp)

Reload local profile previously saved given the login and password given.
- [Boat](#) [testShot](#) ([Shot](#) s)

Test if a boat is touched or sunk by a shot.
- void [setListUser](#) (List< [User](#) > u)

Mutator for list of users.
- void [setListGame](#) (List< [Game](#) > g)

Mutator for list of games.
- [Player](#) [getOtherPLayer](#) ()

Accessor for local player.
- [Player](#) [getLocalPlayerInGame](#) ()

Accessor for the local player in the game.
- boolean [isPlayerOf](#) ([Game](#) game)

To know if the is local player belongs to the game.
- void [gameOver](#) ()

When game is over.
- void [immediateDefeat](#) ()

When immediate defeat.
- void [recordVictory](#) ()

To record victory.

Package Functions

- void [endGame](#) ()

To end the game.
- int [getPlayerPosition](#) ([Player](#) p, [Game](#) g)

Accessor for the player's position in a game (first or second player)
- void [clearData](#) ()

erase the local data, after a disconnection

Private Attributes

- [CDataCom](#) [interfaceDataCom](#)
- [CDataMain](#) [interfaceDataMain](#)
- [CDataTable](#) [interfaceDataTable](#)
- [GuiMainInterface](#) [interfaceMain](#)
- [GuiTableInterface](#) [interfaceTable](#)
- [COMInterface](#) [interfaceCom](#)
- [User](#) [localUser](#)

- [Game](#) localGame
- [Game](#) attendedGame
- [List< User >](#) listUsers
- [Profile](#) localProfile
- [List< Game >](#) listGames
- [Player](#) localPlayer

7.20.1 Detailed Description

Data's controller.

7.20.2 Constructor & Destructor Documentation

7.20.2.1 `data.DataController.DataController ()`

[DataController](#).

7.20.3 Member Function Documentation

7.20.3.1 `void data.DataController.addGameToList (Game g)`

Add a Game to the local list.

Parameters

<code>g</code>	: game to add to the local list
----------------	---------------------------------

7.20.3.2 `void data.DataController.addUserToList (User u)`

Add User to local list.

Parameters

<code>u</code>	: User to add
----------------	---------------

7.20.3.3 `void data.DataController.clearData ()` [package]

erase the local data, after a disconnection

7.20.3.4 `void data.DataController.endGame ()` [package]

To end the game.

7.20.3.5 void data.DataController.gameOver ()

When game is over.

7.20.3.6 Game data.DataController.getAttendedGame ()

Accessor local Game.

Returns

the local Game

7.20.3.7 CDataCom data.DataController.getInterfaceDataCom ()

Accessor for interfaceDataCom.

Returns

interfaceDataCom

7.20.3.8 CDataMain data.DataController.getInterfaceDataMain ()

Accessor for interfaceDataMain.

Returns

interfaceDataMain

7.20.3.9 CDataTable data.DataController.getInterfaceDataTable ()

Accessor for interfaceDataTable.

Returns

interfaceDataTable

7.20.3.10 List<Game> data.DataController.getListGames ()

Get list of Games.

Returns

the list of games

7.20.3.11 List<User> data.DataController.getListUsers ()

Get list of Users.

Returns

the list of users

7.20.3.12 DataUser data.DataController.getLocalDataUser ()

Accessor for local DataUser.

Returns

the local DataUser

7.20.3.13 Game data.DataController.getLocalGame ()

Accessor local Game.

Returns

the local Game

7.20.3.14 Player data.DataController.getLocalPlayer ()

Accessor for local player.

Returns

the local player

7.20.3.15 Player data.DataController.getLocalPlayerInGame ()

Accessor for the local player in the game.

Returns

the player

7.20.3.16 Profile data.DataController.getLocalProfile ()

Accessor for local Profile.

Returns

the local Profile

7.20.3.17 User `data.DataController.getLocalUser ()`

Accessor for the local user.

Returns

local user

7.20.3.18 Player `data.DataController.getOtherPLayer ()`

Accessor for local player.

Returns

the local player

7.20.3.19 int `data.DataController.getPlayerPosition (Player p, Game g)` [`package`]

Accessor for the player's position in a game (first or second player)

Parameters

<i>p</i>	player
<i>g</i>	game

Returns

an integer indicating the player's position

7.20.3.20 GuiTableInterface `data.DataController.getTableInterface ()`

Accessor for table's interface.

Returns

table's interface

7.20.3.21 void `data.DataController.immediateDefeat ()`

When immediate defeat.

7.20.3.22 boolean `data.DataController.isPlayer1 ()`

Enable to know if the local player is the player 1.

Returns

true if the local player is player 1, false otherwise

7.20.3.23 boolean data.DataController.isPlayerOf (Game *game*)

To know if the is local player belongs to the game.

Parameters

<i>game</i>	game to test
-------------	--------------

Returns

a boolean indicating if the is player belongs to the game

7.20.3.24 void data.DataController.recordVictory ()

To record victory.

7.20.3.25 void data.DataController.reloadSavedProfile (String *login*, String *mdp*)

Reload local profile previously saved given the login and password given.

Parameters

<i>login</i>	: login written by User
<i>mdp</i>	: password written by User

7.20.3.26 void data.DataController.removeGameFromList (Game *g*)

Remove the game.

Parameters

<i>g</i>	: game has to be remove
----------	-------------------------

7.20.3.27 void data.DataController.removeUserFromList (User *u*)

Remove a user form local list.

Parameters

<i>u</i>	: User to remove
----------	------------------

7.20.3.28 void data.DataController.setAttendedGame (Game *g*)

Mutator for the attended game.

Parameters

<i>g</i>	attended game
----------	---------------

7.20.3.29 void data.DataController.setInterfaceCom (COMInterface *i*)

Mutator for interfaceCom.

Parameters

<i>i</i>	: new interfaceCom
----------	--------------------

7.20.3.30 void data.DataController.setInterfaceMain (GuiMainInterface *i*)

Mutator for interfaceMain.

Parameters

<i>i</i>	: new interfaceMain
----------	---------------------

7.20.3.31 void data.DataController.setInterfaceTable (GuiTableInterface *i*)

Set the table's interface.

Parameters

<i>i</i>	table's interface
----------	-------------------

7.20.3.32 void data.DataController.setListGame (List< Game > *g*)

Mutator for list of games.

Parameters

<i>g</i>	list of games
----------	---------------

7.20.3.33 void data.DataController.setListUser (List< User > *u*)

Mutator for list of users.

Parameters

<i>u</i>	list of users
----------	---------------

7.20.3.34 void data.DataController.setLocalDataUser (**DataUser** *du*)

Mutator for local DataUser.

Parameters

<i>du</i>	: new DataUser
-----------	----------------

7.20.3.35 void data.DataController.setLocalGame (**Game** *g*)

Mutator local Game.

Parameters

<i>g</i>	: new local Game
----------	------------------

7.20.3.36 void data.DataController.setLocalPlayer (**Player** *p*)

Mutator for local player.

Parameters

<i>p</i>	local player to set
----------	---------------------

7.20.3.37 void data.DataController.setLocalProfile (**Profile** *p*)

Mutator for local Profile.

Parameters

<i>p</i>	: new local Profile
----------	---------------------

7.20.3.38 void data.DataController.setLocalUser (**User** *u*)

Mutator for User.

Parameters

<i>u</i>	: new User
----------	------------

7.20.3.39 Boat data.DataController.testShot (Shot *s*)

Test if a boat is touched or sunk by a shot.

Parameters

<i>s</i>	: shot of the opponent to test
----------	--------------------------------

Returns

a boat if a boat has been sunk

7.20.3.40 void data.DataController.updateGameData (Boolean *ok*, Player *player1*, Player *player2*)

Used by the method setGameJoinResponse of [CDataCom](#).

Parameters

<i>ok</i>	
<i>player1</i>	
<i>player2</i>	

7.20.3.41 void data.DataController.updateGameDataPlaying (Shot *s*, Boat *b*, boolean *forLocalPlayer*)

To update the game's data when players are playing.

Parameters

<i>s</i>	shot
<i>b</i>	boat
<i>forLocalPlayer</i>	if used by the local player or not

7.20.3.42 void data.DataController.updateGameStatus (Game *g*)

Update the status of game and add it into the list of Game.

Parameters

<i>g</i>	: the game which status has been modified
----------	---

7.20.4 Member Data Documentation

7.20.4.1 Game data.DataController.attendedGame [private]

- 7.20.4.2 **COMInterface** `data.DataController.interfaceCom` [private]
- 7.20.4.3 **CDataCom** `data.DataController.interfaceDataCom` [private]
- 7.20.4.4 **CDataMain** `data.DataController.interfaceDataMain` [private]
- 7.20.4.5 **CDataTable** `data.DataController.interfaceDataTable` [private]
- 7.20.4.6 **GuiMainInterface** `data.DataController.interfaceMain` [private]
- 7.20.4.7 **GuiTableInterface** `data.DataController.interfaceTable` [private]
- 7.20.4.8 **List<Game>** `data.DataController.listGames` [private]
- 7.20.4.9 **List<User>** `data.DataController.listUsers` [private]
- 7.20.4.10 **Game** `data.DataController.localGame` [private]
- 7.20.4.11 **Player** `data.DataController.localPlayer` [private]
- 7.20.4.12 **Profile** `data.DataController.localProfile` [private]
- 7.20.4.13 **User** `data.DataController.localUser` [private]

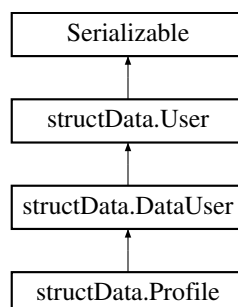
The documentation for this class was generated from the following file:

- `src/main/java/data/`[DataController.java](#)

7.21 structData.DataUser Class Reference

[DataUser](#) class : inherits from [User](#).

Inheritance diagram for structData.DataUser:



Public Member Functions

- [DataUser](#) ([User](#) u)
Constructor by default.
- [DataUser](#) ([User](#) u, String passworddata, List< [ContactGroup](#) > listContactsdata)
Constructor with parameters.
- [DataUser](#) ([DataUser](#) du)
Constructor for the class [DataUser](#).
- String [getPassword](#) ()
Accessor for Password.
- List< [ContactGroup](#) > [getListContacts](#) ()
Accessor for List of contacts.
- void [setPassword](#) (String passworddata)
Mutator for password.
- void [setListContacts](#) (List< [ContactGroup](#) > listContactsdata)
Mutator for List of contacts.
- [DataUser cloneDataUser](#) ([DataUser](#) duserclone)
Clone function to copy a [DataUser](#).

Protected Attributes

- String [password](#)
- List< [ContactGroup](#) > [listContacts](#)

Additional Inherited Members

7.21.1 Detailed Description

[DataUser](#) class : inherits from [User](#).

7.21.2 Constructor & Destructor Documentation

7.21.2.1 structData.DataUser.DataUser ([User](#) u)

Constructor by default.

Parameters

u	: the user of the parent class User . This user has the heritage relationship with the object of the class DataUser
-------------------	---

7.21.2.2 structData.DataUser.DataUser ([User](#) u, String passworddata, List< [ContactGroup](#) > listContactsdata)

Constructor with parameters.

Parameters

<i>u</i>	: the parent user of the new object of class DataUser
<i>passworddata</i>	: the password of the new object of class DataUser
<i>listContactsdata</i>	: the list of contacts of the new object of class DataUser

7.21.2.3 structData.DataUser.DataUser ([DataUser du](#))

Constructor for the class [DataUser](#).

Parameters

<i>du</i>	: a new object of the class DataUser
-----------	--

7.21.3 Member Function Documentation

7.21.3.1 [DataUser](#) structData.DataUser.cloneDataUser ([DataUser duserclone](#))

Clone function to copy a [DataUser](#).

Parameters

<i>duserclone</i>	: the object of the datauser to be cloned
-------------------	---

Returns

the datauser who has called this method to clone all the information of the duserclone

7.21.3.2 [List<ContactGroup>](#) structData.DataUser.getListContacts ()

Accessor for List of contacts.

Returns

the list of contacts of the object of the class [DataUser](#)

7.21.3.3 [String](#) structData.DataUser.getPassword ()

Accessor for Password.

Returns

the password of the object of the class [DataUser](#)

7.21.3.4 [void](#) structData.DataUser.setListContacts ([List< ContactGroup >](#) *listContactsdata*)

Mutator for List of contacts.

Parameters

<code>listContactsdata</code>	: the new value of list of contacts of the object of the class DataUser
-------------------------------	---

7.21.3.5 `void structData.DataUser.setPassword (String passworddata)`

Mutator for password.

Parameters

<code>passworddata</code>	: the new value of password of the object of the class DataUser
---------------------------	---

7.21.4 Member Data Documentation

7.21.4.1 `List<ContactGroup> structData.DataUser.listContacts` [protected]

7.21.4.2 `String structData.DataUser.password` [protected]

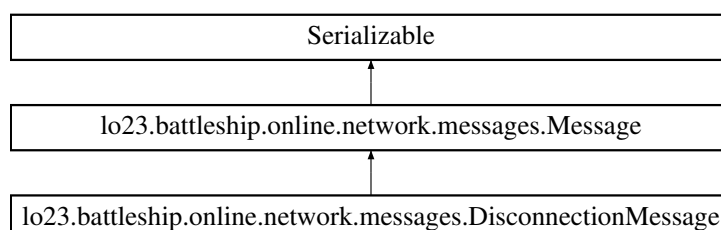
The documentation for this class was generated from the following file:

- `src/main/java/structData/DataUser.java`

7.22 lo23.battleship.online.network.messages.DisconnectionMessage Class Reference

This class implements the message which is sent when when a user disconnects from the application.

Inheritance diagram for `lo23.battleship.online.network.messages.DisconnectionMessage`:



Public Member Functions

- `DisconnectionMessage (User userWhoDisconnect, Game game)`
Allocates a new.
- `String getType ()`
Returns the type of the message.
- `void process (IDataCom IData, InetAddress senderAddress)`
Updates the network state and the list of connected users using data package interface and the [NetworkController](#) : removes the disconnected user from those 2 collections.

Package Attributes

- [User](#) `user`
- [Game](#) `game`

7.22.1 Detailed Description

This class implements the message which is sent when when a user disconnects from the application.

The user who wants to disconnect sends this message to all connected users.

This class extends the abstract [Message](#) and implements the two abstract methods: `getType` and `process`

See also

[Message](#)

Author

COM Module

7.22.2 Constructor & Destructor Documentation

7.22.2.1 `io23.battleship.online.network.messages.DisconnectionMessage.DisconnectionMessage (User userWhoDisconnect, Game game)`

Allocates a new.

`DisconnectionMessage`

object.

Parameters

<code><i>userWhoDisconnect</i></code>	
---------------------------------------	--

7.22.3 Member Function Documentation

7.22.3.1 `String io23.battleship.online.network.messages.DisconnectionMessage.getType ()`

Returns the type of the message.

Implementation of an abstract method.

Returns

type :
`String`

7.22.3.2 void `lo23.battleship.online.network.messages.DisconnectionMessage.process (IDataCom IData, InetAddress senderAddress)`

Updates the network state and the list of connected users using data package interface and the [NetworkController](#) : removes the disconnected user from those 2 collections.

Parameters

<i>IData</i>	: IDataCom instance of IDataCom interface.
<i>senderAddress</i>	: InetAddress sender's IP address

7.22.4 Member Data Documentation

7.22.4.1 **Game** `lo23.battleship.online.network.messages.DisconnectionMessage.game` [package]

7.22.4.2 **User** `lo23.battleship.online.network.messages.DisconnectionMessage.user` [package]

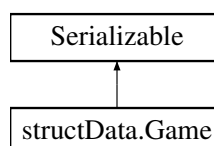
The documentation for this class was generated from the following file:

- `src/main/java/lo23/battleship/online/network/messages/DisconnectionMessage.java`

7.23 structData.Game Class Reference

[Game](#) is a class for the user's game.

Inheritance diagram for structData.Game:



Public Member Functions

- [Game](#) ([Profile](#) p)
Default constructor.
- [Game](#) (Boolean newClassicType, String newName, Boolean newHumanOpponent, Integer newTimePerShot, Integer newTimeToPlaceBoats, Boolean newSpectator, Boolean newSpectatorChat, [Profile](#) p)
Constructor with all parameters.

- [Game](#) (Boolean newClassicType, String newName, Boolean newHumanOpponent, Integer newTimePerShot, Integer newTimeToPlaceBoats, Boolean newSpectator, Boolean newSpectatorChat, [Profile](#) p, HashSet IS, [Player](#) p1, [Player](#) p2, Boolean p1Start, ArrayList IMsg)
 - Second constructor with all parameters.*
- [Game](#) ([Game](#) g)
 - Constructor with a [Game](#).*
- [Game](#) clone ([Game](#) g)
 - Method to copy a [Game](#) : Clone method.*
- UUID [getIdGame](#) ()
 - Accessor for idGame.*
- void [setStatus](#) ([StatusGame](#) status)
 - Mutator for status.*
- [StatusGame](#) [getStatus](#) ()
 - Accessor for status.*
- Boolean [getClassicType](#) ()
 - Accessor for classicType.*
- String [getName](#) ()
 - Accessor for name.*
- Boolean [getHumanOpponent](#) ()
 - Accessor for humanOpponent.*
- int [getTimePerShot](#) ()
 - Accessor for timePerShot.*
- Boolean [getSpectator](#) ()
 - Accessor for spectator.*
- Boolean [getSpectatorChat](#) ()
 - Accessor for spectatorChat.*
- void [setListMessages](#) (ArrayList IMsg)
 - Mutator for the list of messages.*
- ArrayList [getListMessages](#) ()
 - Accessor for the list of messages.*
- void [addMessage](#) ([ChatMessage](#) msg)
 - Add a message to the list of message.*
- void [setListSpectators](#) (HashSet< [User](#) > IS)
 - Mutator for the list of spectator.*
- HashSet [getListSpectators](#) ()
 - Accessor for the list of spectator.*
- void [addSpectators](#) ([User](#) spectator)
 - Add a spectator to the list of spectator.*
- [Player](#) [getPlayer1](#) ()
 - Accessor for the player 1.*
- [Player](#) [getPlayer2](#) ()
 - Accessor for the player 2.*
- Boolean [getPlayer1Start](#) ()
 - Accessor for player1Start.*
- void [setPlayer1](#) ([Player](#) p1)
 - Mutator for player 1.*
- void [setPlayer2](#) ([Player](#) p2)
 - Mutator for player 2.*
- Boolean [doesProfileBelongToGame](#) ([Profile](#) p)
 - Check if a profile belongs to a [Game](#).*
- int [getTimeToPlaceBoats](#) ()
 - Accessor for timeToPlaceBoats.*

Static Package Attributes

- static final long [serialVersionUID](#) = 4L

Private Attributes

- UUID [idGame](#)
- [StatusGame](#) [status](#)
- Boolean [classicType](#)
- String [name](#)
- Boolean [humanOpponent](#)
- Integer [timePerShot](#)
- Integer [timeToPlaceBoats](#)
- Boolean [spectator](#)
- Boolean [spectatorChat](#)
- HashSet< [User](#) > [listSpectators](#)
- [Player](#) [player1](#)
- [Player](#) [player2](#)
- Boolean [player1Start](#)
- ArrayList [listMessages](#)

7.23.1 Detailed Description

[Game](#) is a class for the user's game.

7.23.2 Constructor & Destructor Documentation

7.23.2.1 `structData.Game.Game (Profile p)`

Default constructor.

Parameters

<i>p</i>	: Profile of the user creating the Game
----------	---

7.23.2.2 `structData.Game.Game (Boolean newClassicType, String newName, Boolean newHumanOpponent, Integer newTimePerShot, Integer newTimeToPlaceBoats, Boolean newSpectator, Boolean newSpectatorChat, Profile p)`

Constructor with all parameters.

Parameters

<i>newClassicType</i>	the game type
<i>newName</i>	the game's name
<i>newHumanOpponent</i>	a boolean equal to 1 if the game is between two players and 0 if it is against a bot
<i>newTimePerShot</i>	time per shot
<i>newTimeToPlaceBoats</i>	

Parameters

<i>newSpectator</i>	a boolean equal to 1 if spectators are allowed
<i>newSpectatorChat</i>	a boolean equal to 1 if chat is allowed
<i>p</i>	a Profile

7.23.2.3 `structData.Game.Game (Boolean newClassicType, String newName, Boolean newHumanOpponent, Integer newTimePerShot, Integer newTimeToPlaceBoats, Boolean newSpectator, Boolean newSpectatorChat, Profile p, HashSet IS, Player p1, Player p2, Boolean p1Start, ArrayList IMsg)`

Second constructor with all parameters.

Parameters

<i>newClassicType</i>	the game type
<i>newName</i>	the game's name
<i>newHumanOpponent</i>	a boolean equal to 1 if the game is between two players and 0 if it is against a bot
<i>newTimePerShot</i>	time per shot
<i>newTimeToPlaceBoats</i>	
<i>newSpectator</i>	a boolean equal to 1 if spectators are allowed
<i>newSpectatorChat</i>	a boolean equal to 1 if chat is allowed
<i>p</i>	a Profile
<i>IS</i>	a Hashset
<i>p1</i>	a Player
<i>p2</i>	a Player
<i>p1Start</i>	a Boolean
<i>IMsg</i>	an ArrayList

7.23.2.4 `structData.Game.Game (Game g)`

Constructor with a [Game](#).

Parameters

<i>g</i>	a Game
----------	------------------------

7.23.3 Member Function Documentation

7.23.3.1 `void structData.Game.addMessage (ChatMessage msg)`

Add a message to the list of message.

Parameters

<i>msg</i>	the message to add
------------	--------------------

7.23.3.2 void structData.Game.addSpectators (User *spectator*)

Add a spectator to the list of spectator.

Parameters

<i>spectator</i>	the user to add
------------------	-----------------

7.23.3.3 Game structData.Game.clone (Game *g*)

Method to copy a [Game](#) : Clone method.

Parameters

<i>g</i>	a Game to copy
----------	--------------------------------

Returns

a copied [Game](#)

7.23.3.4 Boolean structData.Game.doesProfileBelongToGame (Profile *p*)

Check if a profile belongs to a [Game](#).

Parameters

<i>p</i>	: a profile
----------	-------------

Returns

a boolean if the profile belongs to one of the [Game](#)'s Players

7.23.3.5 Boolean structData.Game.getClassicType ()

Accessor for classicType.

Returns

the game's classicType as a boolean

7.23.3.6 Boolean structData.Game.getHumanOpponent ()

Accessor for humanOpponent.

Returns

the game's humanOpponent as a boolean

7.23.3.7 UUID structData.Game.getIdGame ()

Accessor for idGame.

Returns

the [Game](#)'s id

7.23.3.8 ArrayList structData.Game.getListMessages ()

Accessor for the list of messages.

Returns

a list of message as an ArrayList

7.23.3.9 HashSet structData.Game.getListSpectators ()

Accessor for the list of spectator.

Returns

a list of spectator as an HashSet

7.23.3.10 String structData.Game.getName ()

Accessor for name.

Returns

the game's name as a string

7.23.3.11 Player structData.Game.getPlayer1 ()

Accessor for the player 1.

Returns

player 1 as a [Player](#)

7.23.3.12 Boolean structData.Game.getPlayer1Start ()

Accessor for player1 Start.

Returns

player1Start as a Boolean

7.23.3.13 **Player** structData.Game.getPlayer2 ()

Accessor for the player 2.

Returns

player 2 as a [Player](#)

7.23.3.14 **Boolean** structData.Game.getSpectator ()

Accessor for spectator.

Returns

the game's spectator as a boolean

7.23.3.15 **Boolean** structData.Game.getSpectatorChat ()

Accessor for spectatorChat.

Returns

the game's spectatorChat as a boolean

7.23.3.16 **StatusGame** structData.Game.getStatus ()

Accessor for status.

Returns

the game's status as a [StatusGame](#)

7.23.3.17 **int** structData.Game.getTimePerShot ()

Accessor for timePerShot.

Returns

the game's timePerShot as an integer

7.23.3.18 **int** structData.Game.getTimeToPlaceBoats ()

Accessor for timeToPlaceBoats.

Returns

the time to place boats

7.23.3.19 **void** structData.Game.setListMessages (*ArrayList IMessage*)

Mutator for the list of messages.

Parameters

<i>IMsg</i>	a HashSet
-------------	-----------

7.23.3.20 void structData.Game.setListSpectators (HashSet< User > *IS*)

Mutator for the list of spectator.

Parameters

<i>IS</i>	a HashSet
-----------	-----------

7.23.3.21 void structData.Game.setPlayer1 (Player *p1*)

Mutator for player 1.

Parameters

<i>p1</i>	the new player 1
-----------	------------------

7.23.3.22 void structData.Game.setPlayer2 (Player *p2*)

Mutator for player 2.

Parameters

<i>p2</i>	: the new player 2
-----------	--------------------

7.23.3.23 void structData.Game.setStatus (StatusGame *status*)

Mutator for status.

Parameters

<i>status</i>	the game's new status
---------------	-----------------------

7.23.4 Member Data Documentation

7.23.4.1 Boolean structData.Game.classicType [private]

7.23.4.2 Boolean structData.Game.humanOpponent [private]

- 7.23.4.3 `UUID structData.Game.idGame` [private]
- 7.23.4.4 `ArrayList structData.Game.listMessages` [private]
- 7.23.4.5 `HashSet<User> structData.Game.listSpectators` [private]
- 7.23.4.6 `String structData.Game.name` [private]
- 7.23.4.7 `Player structData.Game.player1` [private]
- 7.23.4.8 `Boolean structData.Game.player1Start` [private]
- 7.23.4.9 `Player structData.Game.player2` [private]
- 7.23.4.10 `final long structData.Game.serialVersionUID = 4L` [static], [package]
- 7.23.4.11 `Boolean structData.Game.spectator` [private]
- 7.23.4.12 `Boolean structData.Game.spectatorChat` [private]
- 7.23.4.13 `StatusGame structData.Game.status` [private]
- 7.23.4.14 `Integer structData.Game.timePerShot` [private]
- 7.23.4.15 `Integer structData.Game.timeToPlaceBoats` [private]

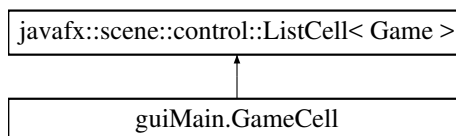
The documentation for this class was generated from the following file:

- [src/main/java/structData/Game.java](#)

7.24 guiMain.GameCell Class Reference

[GameCell](#), descendant class `ListCell<Game>`, display the game name and offers the possibility to join or watch a game.

Inheritance diagram for `guiMain.GameCell`:



Public Member Functions

- [GameCell](#) ([menuController](#) c)
Class constructor.

Protected Member Functions

- void `updateItem` (final `Game` game, boolean empty)
Update the content of an item : set a style according to the parameters of a game.

Package Attributes

- `HBox hbox` = new `HBox()`
- `HBox joinButtonBox` = new `HBox()`
- `HBox lookButtonBox` = new `HBox()`
- `Label title` = new `Label()`
- `Label type` = new `Label()`
- `ImageView robot` = new `ImageView()`
- `Button joinButton` = new `Button("")`
- `Button lookButton` = new `Button("")`
- `menuController controller`

7.24.1 Detailed Description

`GameCell`, descendant class `ListCell<Game>`, display the game name and offers the possibility to join or watch a game.

Author

IHM-MAIN Module

7.24.2 Constructor & Destructor Documentation

7.24.2.1 guiMain.GameCell.GameCell (menuController c)

Class constructor.

Parameters

<code>c</code>	is the menu controller where the game is display
----------------	--

7.24.3 Member Function Documentation

7.24.3.1 void guiMain.GameCell.updateItem (final Game game, boolean empty) [protected]

Update the content of an item : set a style according to the parameters of a game.

Parameters

<code>game</code>	: the game wich would be display
<code>empty</code>	: true if the cell is empty

7.24.4 Member Data Documentation

7.24.4.1 `menuController guiMain.GameCell.controller` [package]

7.24.4.2 `HBox guiMain.GameCell.hbox = new HBox()` [package]

7.24.4.3 `Button guiMain.GameCell.joinButton = new Button("")` [package]

7.24.4.4 `HBox guiMain.GameCell.joinButtonBox = new HBox()` [package]

7.24.4.5 `Button guiMain.GameCell.lookButton = new Button("")` [package]

7.24.4.6 `HBox guiMain.GameCell.lookButtonBox = new HBox()` [package]

7.24.4.7 `ImageView guiMain.GameCell.robot = new ImageView()` [package]

7.24.4.8 `Label guiMain.GameCell.title = new Label()` [package]

7.24.4.9 `Label guiMain.GameCell.type = new Label()` [package]

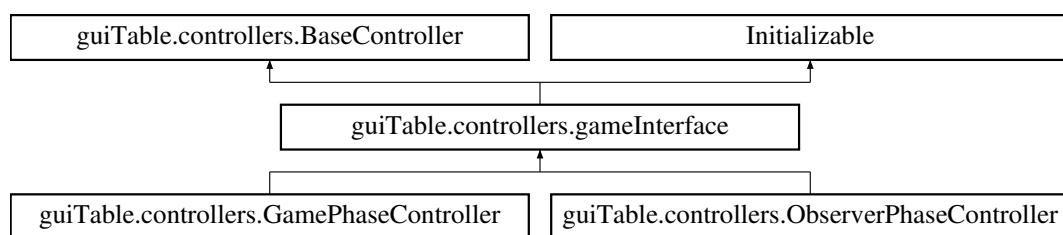
The documentation for this class was generated from the following file:

- [src/main/java/guiMain/GameCell.java](#)

7.25 guiTable.controllers.gameInterface Class Reference

[gameInterface](#)

Inheritance diagram for `guiTable.controllers.gameInterface`:



Public Member Functions

- void [showVictory](#) ()
Shows victory message.
- void [showDefeat](#) ()
Shows defeat message.
- void [logMsg](#) (String msg)
log message into interface.
- void [logYesNoMsg](#) (String msg)
log yesNoMessage into interface.

Protected Member Functions

- void [placeShotTo](#) ([Shot](#) shot, [GridPane](#) grid)
- void [sunkABoat](#) ([GridPane](#) grid, [Boat](#) boat)

Protected Attributes

- Button [noButton](#)
- Button [exitButton](#)
- Button [yesButton](#)
- Text [messageTextContainer](#)
- Pane [messageContainer](#)
- AnchorPane [chatPane](#)
- [GuiTableController](#) [tableController](#)
- final String [STYLE_MY_TURN](#) = "-fx-background-color: #FFFFFF;"
- final String [STYLE_OTHER_TURN](#) = "-fx-background-color: #EEEEEE;"
- String [MY_TURN_MSG](#) = "A votre tour de jouer, cliquer sur une case puis sur le bouton valider"
- String [OTHER_TURN_MSG](#) = "Au tour de l'adversaire de jouer, merci de patienter"
- String [EXIT_GAME_MSG](#) = "Voulez-vous vraiment quitter la partie ?"
- String [VICTORY_MSG](#) = "Victoire !"
- String [DEFEAT_MSG](#) = "Defaite !"
- boolean [waitExit](#) = false
- String [waitMsg](#)

Package Functions

- void [yesClicked](#) ([ActionEvent](#) event)
Validate end of game.

Private Attributes

- AnchorPane [anchorPane](#)

Additional Inherited Members

7.25.1 Detailed Description

[gameInterface](#)

7.25.2 Member Function Documentation

7.25.2.1 void guiTable.controllers.gameInterface.logMsg (String msg)

log message into interface.

Parameters

<i>msg</i>	message to be displayed
------------	-------------------------

7.25.2.2 void guiTable.controllers.gameInterface.logYesNoMsg (String *msg*)

log yesNoMessage into interface.

Parameters

<i>msg</i>	message to be displayed
------------	-------------------------

7.25.2.3 void guiTable.controllers.gameInterface.placeShotTo (Shot *shot*, GridPane *grid*) [protected]

7.25.2.4 void guiTable.controllers.gameInterface.showDefeat ()

Shows defeat message.

7.25.2.5 void guiTable.controllers.gameInterface.showVictory ()

Shows victory message.

7.25.2.6 void guiTable.controllers.gameInterface.sunkABoat (GridPane *grid*, Boat *boat*) [protected]

7.25.2.7 void guiTable.controllers.gameInterface.yesClicked (ActionEvent *event*) [package]

Validate end of game.

7.25.3 Member Data Documentation

7.25.3.1 AnchorPane guiTable.controllers.gameInterface.anchorPane [private]

7.25.3.2 AnchorPane guiTable.controllers.gameInterface.chatPane [protected]

7.25.3.3 String guiTable.controllers.gameInterface.DEFEAT_MSG = "Defaite !" [protected]

7.25.3.4 String guiTable.controllers.gameInterface.EXIT_GAME_MSG = "Voulez-vous vraiment quitter la partie ?"
[protected]

7.25.3.5 Button guiTable.controllers.gameInterface.exitButton [protected]

7.25.3.6 Pane guiTable.controllers.gameInterface.messageContainer [protected]

- 7.25.3.7 Text guiTable.controllers.gameInterface.messageTextContainer [protected]
- 7.25.3.8 String guiTable.controllers.gameInterface.MY_TURN_MSG = "A votre tour de jouer, cliquer sur une case puis sur le bouton valider" [protected]
- 7.25.3.9 Button guiTable.controllers.gameInterface.noButton [protected]
- 7.25.3.10 String guiTable.controllers.gameInterface.OTHER_TURN_MSG = "Au tour de l'adversaire de jouer, merci de patienter" [protected]
- 7.25.3.11 final String guiTable.controllers.gameInterface.STYLE_MY_TURN = "-fx-background-color: #FFFFFF;" [protected]
- 7.25.3.12 final String guiTable.controllers.gameInterface.STYLE_OTHER_TURN = "-fx-background-color: #EEEEEE;" [protected]
- 7.25.3.13 GuiTableController guiTable.controllers.gameInterface.tableController [protected]
- 7.25.3.14 String guiTable.controllers.gameInterface.VICTORY_MSG = "Victoire !" [protected]
- 7.25.3.15 boolean guiTable.controllers.gameInterface.waitExit = false [protected]
- 7.25.3.16 String guiTable.controllers.gameInterface.waitMsg [protected]
- 7.25.3.17 Button guiTable.controllers.gameInterface.yesButton [protected]

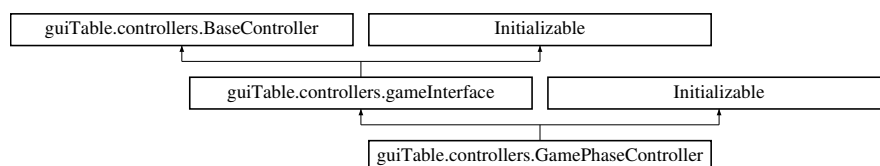
The documentation for this class was generated from the following file:

- src/main/java/guiTable/controllers/[gameInterface.java](#)

7.26 guiTable.controllers.GamePhaseController Class Reference

[GamePhaseController](#).

Inheritance diagram for guiTable.controllers.GamePhaseController:



Public Member Functions

- void [initialize](#) (URL location, ResourceBundle resources)
- void [showVictory](#) ()
- void [showDefeat](#) ()
- void [setMyTurn](#) (Boolean [myTurn](#))
Changes the colors of the boards according to the turns of the players.
- void [addShot](#) ([Shot](#) shot)
Adds a shot on the board.
- void [sunkBoat](#) ([Boat](#) boat)
Sunk a boat.
- void [addOpponentShot](#) ([Shot](#) opponentShot)
Add the opponent shot.
- void [sunkMyBoat](#) ([Boat](#) boat)
Sunk my boat.
- void [setMyBoats](#) (List< [Boat](#) > boats)
Set my boats.
- void [setRoundTime](#) ()
Set the round time.
- void [exitGame](#) ()
- [AnchorPane](#) [getChatPane](#) ()
Get the chat pane.

Public Attributes

- [LocalTime](#) [time](#)
- [LocalTime](#) [timePerShot](#)

Protected Member Functions

- [EventHandler](#)< [MouseEvent](#) > [onClickCase](#) ()
Function dispatched when the user clicks over a case in te boards.
- void [validateShot](#) ()
Sends the shot to the user's opponent and observers.
- void [removeSelectedCase](#) ()
Remove the selected case.
- void [noClicked](#) ()
Cancel end of game.
- void [timelsOver](#) ()
Notify that the time is over.

Private Attributes

- [GridPane](#) [table](#)
- [GridPane](#) [myTable](#)
- [CaseDrawing](#) [selectedCase](#)
- [Label](#) [gameState](#)
- [Rectangle](#) [messageMask](#)
- [Button](#) [valider](#)
- [Label](#) [timerLabel](#)
- Boolean [myTurn](#)
- [Timeline](#) [timeline](#)

Additional Inherited Members

7.26.1 Detailed Description

[GamePhaseController](#).

7.26.2 Member Function Documentation

7.26.2.1 void guiTable.controllers.GamePhaseController.addOpponentShot (Shot *opponentShot*)

Add the opponent shot.

Parameters

<i>opponentShot</i>	
---------------------	--

7.26.2.2 void guiTable.controllers.GamePhaseController.addShot (Shot *shot*)

Adds a shot on the board.

Parameters

<i>shot</i>	
-------------	--

7.26.2.3 void guiTable.controllers.GamePhaseController.exitGame ()

7.26.2.4 AnchorPane guiTable.controllers.GamePhaseController.getChatPane ()

Get the chat pane.

Returns

the chatPane

7.26.2.5 void guiTable.controllers.GamePhaseController.initialize (URL *location*, ResourceBundle *resources*)

7.26.2.6 void guiTable.controllers.GamePhaseController.noClicked () [protected]

Cancel end of game.

7.26.2.7 `EventHandler<MouseEvent> guiTable.controllers.GamePhaseController.onClickCase ()` [protected]

Function dispatched when the user clicks over a case in te boards.

Returns

the EventHandler of this situation

7.26.2.8 `void guiTable.controllers.GamePhaseController.removeSelectedCase ()` [protected]

Remove the selected case.

7.26.2.9 `void guiTable.controllers.GamePhaseController.setMyBoats (List< Boat > boats)`

Set my boats.

Parameters

<i>boats</i>	
--------------	--

7.26.2.10 `void guiTable.controllers.GamePhaseController.setMyTurn (Boolean myTurn)`

Changes the colors of the boards according to the turns of the players.

Parameters

<i>myTurn</i>	
---------------	--

7.26.2.11 `void guiTable.controllers.GamePhaseController.setRoundTime ()`

Set the round time.

7.26.2.12 `void guiTable.controllers.GamePhaseController.showDefeat ()`

7.26.2.13 `void guiTable.controllers.GamePhaseController.showVictory ()`

7.26.2.14 `void guiTable.controllers.GamePhaseController.sunckBoat (Boat boat)`

Sunk a boat.

Parameters

<i>boat</i>	
-------------	--

7.26.2.15 void guiTable.controllers.GamePhaseController.sunkMyBoat (*Boat boat*)

Sunk my boat.

Parameters

<i>boat</i>	
-------------	--

7.26.2.16 void guiTable.controllers.GamePhaseController.timeIsOver () [protected]

Notify that the time is over.

7.26.2.17 void guiTable.controllers.GamePhaseController.validateShot () [protected]

Sends the shot to the user's opponent and observers.

7.26.3 Member Data Documentation

7.26.3.1 Label guiTable.controllers.GamePhaseController.gameState [private]

7.26.3.2 Rectangle guiTable.controllers.GamePhaseController.messageMask [private]

7.26.3.3 GridPane guiTable.controllers.GamePhaseController.myTable [private]

7.26.3.4 Boolean guiTable.controllers.GamePhaseController.myTurn [private]

7.26.3.5 CaseDrawing guiTable.controllers.GamePhaseController.selectedCase [private]

7.26.3.6 GridPane guiTable.controllers.GamePhaseController.table [private]

7.26.3.7 LocalTime guiTable.controllers.GamePhaseController.time

7.26.3.8 Timeline guiTable.controllers.GamePhaseController.timeline [private]

7.26.3.9 LocalTime guiTable.controllers.GamePhaseController.timePerShot

7.26.3.10 Label guiTable.controllers.GamePhaseController.timerLabel [private]

7.26.3.11 Button guiTable.controllers.GamePhaseController.validator [private]

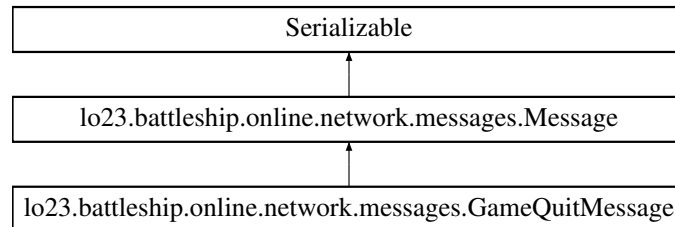
The documentation for this class was generated from the following file:

- src/main/java/guiTable/controllers/[GamePhaseController.java](#)

7.27 lo23.battleship.online.network.messages.GameQuitMessage Class Reference

This class implements the message sent when a game is removed from the network (and the list of games) by his creator(user) or when the game is over.

Inheritance diagram for lo23.battleship.online.network.messages.GameQuitMessage:



Public Member Functions

- [GameQuitMessage](#) ([Game](#) game)
- String [getType](#) ()
Returns the type of the message.
- void [process](#) ([IDataCom](#) IData, [InetAddress](#) senderAddress)
Remove the game `game` from the list of games using data package interface `IData` and the [NetworkController](#) : removes the disconnected user from those 2 collections.

Private Attributes

- [Game](#) game

Additional Inherited Members

7.27.1 Detailed Description

This class implements the message sent when a game is removed from the network (and the list of games) by his creator(user) or when the game is over.

This class extends the abstract [Message](#) and implements the two abstract methods: `getType` and `process`

See also

[Message](#)

Author

COM Module

7.27.2 Constructor & Destructor Documentation

7.27.2.1 `lo23.battleship.online.network.messages.GameQuitMessage.GameQuitMessage (Game game)`

7.27.3 Member Function Documentation

7.27.3.1 `String lo23.battleship.online.network.messages.GameQuitMessage.getType ()`

Returns the type of the message.

Implementation of an abstract method.

Returns

type :

`String`

7.27.3.2 `void lo23.battleship.online.network.messages.GameQuitMessage.process (IDataCom IData, InetAddress senderAddress)`

Remove the game `game` from the list of games using data package interface `IData` and the [NetworkController](#) : removes the disconnected user from those 2 collections.

Parameters

<i>IData</i>	: <code>IDataCom</code> instance of <code>IDataCom</code> interface.
<i>senderAddress</i>	: <code>InetAddress</code> sender's IP address

7.27.4 Member Data Documentation

7.27.4.1 `Game lo23.battleship.online.network.messages.GameQuitMessage.game` `[private]`

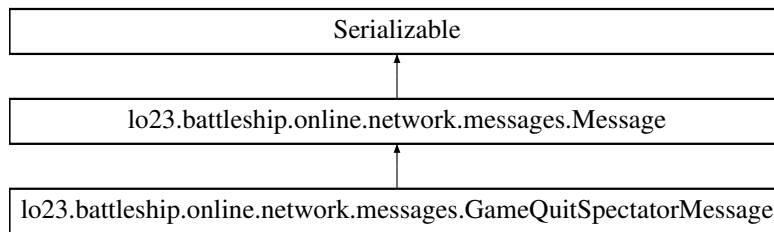
The documentation for this class was generated from the following file:

- `src/main/java/lo23/battleship/online/network/messages/GameQuitMessage.java`

7.28 `lo23.battleship.online.network.messages.GameQuitSpectatorMessage` Class Reference

This class implements the message which is sent when a spectator quit a game he is watching.

Inheritance diagram for `lo23.battleship.online.network.messages.GameQuitSpectatorMessage`:



Public Member Functions

- [GameQuitSpectatorMessage](#) ([Game](#) g, [User](#) s)
- [String](#) [getType](#) ()
Returns the type of the message.
- [void](#) [process](#) ([IDataCom](#) IData, [InetAddress](#) senderAddress)
execute the process associated to the GameQuitSpectator message It notifies the data interface of users who received this message that the spectator spec has quit watching the game game

Private Attributes

- [Game](#) game
- [User](#) spec

Additional Inherited Members

7.28.1 Detailed Description

This class implements the message which is sent when a spectator quit a game he is watching.

This class extends the abstract [Message](#) and implements the two abstract methods: `getType` and `process`

See also

[Message](#)

Author

COM Module

7.28.2 Constructor & Destructor Documentation

- 7.28.2.1 `lo23.battleship.online.network.messages.GameQuitSpectatorMessage.GameQuitSpectatorMessage (Game g, User s)`

7.28.3 Member Function Documentation

- 7.28.3.1 `String lo23.battleship.online.network.messages.GameQuitSpectatorMessage.getType ()`

Returns the type of the message.

Implementation of an abstract method.

Returns

type :
`String`

7.28.3.2 void lo23.battleship.online.network.messages.GameQuitSpectatorMessage.process (IDataCom *IData*, InetAddress *senderAddress*)

execute the process associated to the GameQuitSpectator message It notifies the *data* interface of users who received this message that the spectator *spec* has quit watching the game *game*

Parameters

<i>IData</i>	: IDataCom instance of IDataCom interface
<i>senderAddress</i>	: InetAddress IP address of the sender

7.28.4 Member Data Documentation

7.28.4.1 Game lo23.battleship.online.network.messages.GameQuitSpectatorMessage.game [private]

7.28.4.2 User lo23.battleship.online.network.messages.GameQuitSpectatorMessage.spec [private]

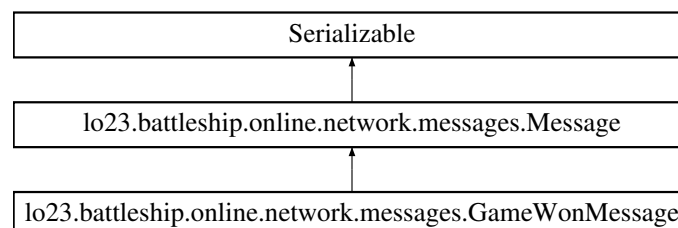
The documentation for this class was generated from the following file:

- src/main/java/lo23/battleship/online/network/messages/[GameQuitSpectatorMessage.java](#)

7.29 lo23.battleship.online.network.messages.GameWonMessage Class Reference

This class implements the message which is sent when a game is over with a victory of one of the players.

Inheritance diagram for lo23.battleship.online.network.messages.GameWonMessage:



Public Member Functions

- [GameWonMessage](#) (Player p)
- String [getType](#) ()
Returns the type of the message.
- void [process](#) (IDataCom *IData*, InetAddress *senderAddress*)
Notifies the other player that he has won and update the data corresponding the game using the data package interface IData

Package Attributes

- [Player player](#)

7.29.1 Detailed Description

This class implements the message which is sent when a game is over with a victory of one of the players.

The loser sends this message to the winner.

This class extends the abstract [Message](#) and implements the two abstract methods: `getType` and `process`

See also

[Message](#)

Author

COM Module

7.29.2 Constructor & Destructor Documentation

7.29.2.1 `Io23.battleship.online.network.messages.GameWonMessage.GameWonMessage (Player p)`

7.29.3 Member Function Documentation

7.29.3.1 `String Io23.battleship.online.network.messages.GameWonMessage.getType ()`

Returns the type of the message.

Implementation of an abstract method.

Returns

type :

`String`

7.29.3.2 `void Io23.battleship.online.network.messages.GameWonMessage.process (IDataCom IData, InetAddress senderAddress)`

Notifies the other player that he has won and update the data corresponding the game using the data package interface `IData`

Parameters

<i>IData</i>	: IDataCom instance of IDataCom interface.
<i>senderAddress</i>	: InetAddress sender's IP address

7.29.4 Member Data Documentation

7.29.4.1 Player lo23.battleship.online.network.messages.GameWonMessage.player [package]

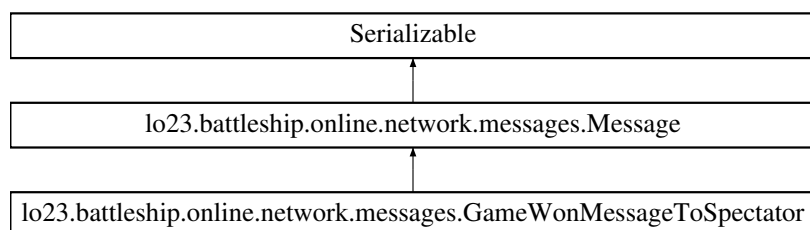
The documentation for this class was generated from the following file:

- src/main/java/lo23/battleship/online/network/messages/[GameWonMessage.java](#)

7.30 lo23.battleship.online.network.messages.GameWonMessageToSpectator Class Reference

This class implements the message which is sent when a game is over with a victory of one of the players.

Inheritance diagram for lo23.battleship.online.network.messages.GameWonMessageToSpectator:



Public Member Functions

- [GameWonMessageToSpectator](#) ([Player](#) p)
- String [getType](#) ()
Returns the type of the message.
- void [process](#) ([IDataCom](#) IData, InetAddress senderAddress)
Notifies the spectators of the winner of the game and update the data corresponding the game using the data package interface IData

Package Attributes

- [Player](#) player

7.30.1 Detailed Description

This class implements the message which is sent when a game is over with a victory of one of the players.

The loser sends this message to the spectators.

This class extends the abstract [Message](#) and implements the two abstract methods: `getType` and `process`

See also

[Message](#)

Author

COM Module

7.30.2 Constructor & Destructor Documentation

7.30.2.1 `Io23.battleship.online.network.messages.GameWonMessageToSpectator.GameWonMessageToSpectator (Player p)`

7.30.3 Member Function Documentation

7.30.3.1 `String Io23.battleship.online.network.messages.GameWonMessageToSpectator.getType ()`

Returns the type of the message.

Implementation of an abstract method.

Returns

type :

`String`

7.30.3.2 `void Io23.battleship.online.network.messages.GameWonMessageToSpectator.process (IDataCom IData, InetAddress senderAddress)`

Notifies the spectators of the winner of the game and update the data corresponding the game using the data package interface `IData`

Parameters

<i>IData</i>	: <code>IDataCom</code> instance of <code>IDataCom</code> interface.
<i>senderAddress</i>	: <code>InetAddress</code> sender's IP address

7.30.4 Member Data Documentation

7.30.4.1 Player `lo23.battleship.online.network.messages.GameWonMessageToSpectator.player` [package]

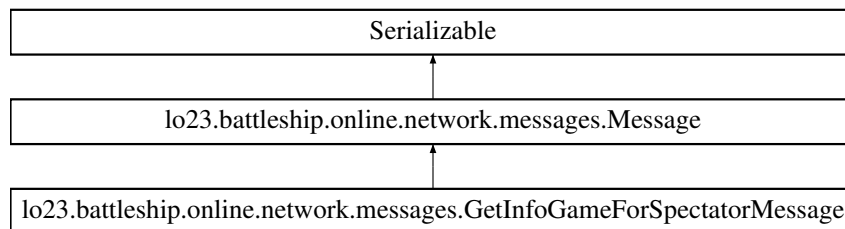
The documentation for this class was generated from the following file:

- `src/main/java/lo23/battleship/online/network/messages/GameWonMessageToSpectator.java`

7.31 `lo23.battleship.online.network.messages.GetInfoGameForSpectatorMessage` Class Reference

This class implements the message which is sent when when a user wants to watch a game.

Inheritance diagram for `lo23.battleship.online.network.messages.GetInfoGameForSpectatorMessage`:



Public Member Functions

- `GetInfoGameForSpectatorMessage` (`Player` p, `User` s)
- `String getType` ()
Returns the type of the message.
- `void process` (`IDataCom` IData, `InetAddress` senderAddress)
Adds the user as spectator to the game (if possible) using the data package interface `IData`

Package Attributes

- `Player` player
- `User` spec

7.31.1 Detailed Description

This class implements the message which is sent when when a user wants to watch a game.

The user sends this message to the creator of the game

This class extends the abstract `Message` and implements the two abstract methods: `getType` and `process`

See also

[Message](#)

Author

COM Module

7.31.2 Constructor & Destructor Documentation

7.31.2.1 `lo23.battleship.online.network.messages.GetInfoGameForSpectatorMessage.GetInfoGameForSpectatorMessage (Player p, User s)`

7.31.3 Member Function Documentation

7.31.3.1 `String lo23.battleship.online.network.messages.GetInfoGameForSpectatorMessage.getType ()`

Returns the type of the message.

Implementation of an abstract method.

Returns

type :
String

7.31.3.2 `void lo23.battleship.online.network.messages.GetInfoGameForSpectatorMessage.process (IDataCom IData, InetAddress senderAddress)`

Adds the user as spectator to the game (if possible) using the data package interface `IData`

Parameters

<i>IData</i>	: IDataCom instance of IDataCom interface.
<i>senderAddress</i>	: InetAddress sender's IP address

7.31.4 Member Data Documentation

7.31.4.1 **Player** `lo23.battleship.online.network.messages.GetInfoGameForSpectatorMessage.player` [package]

7.31.4.2 **User** `lo23.battleship.online.network.messages.GetInfoGameForSpectatorMessage.spec` [package]

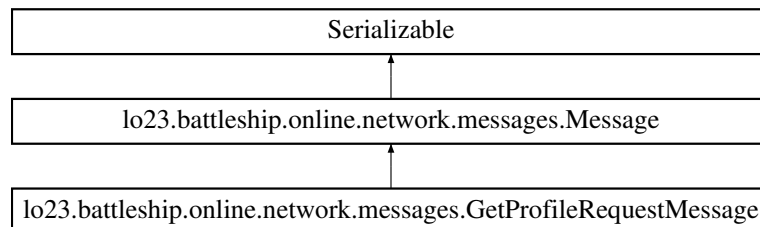
The documentation for this class was generated from the following file:

- `src/main/java/lo23/battleship/online/network/messages/GetInfoGameForSpectatorMessage.java`

7.32 `lo23.battleship.online.network.messages.GetProfileRequestMessage` Class Reference

This class implements the message which is sent when when a user wants to see the profile(including statistics) of an other user.

Inheritance diagram for `lo23.battleship.online.network.messages.GetProfileRequestMessage`:



Public Member Functions

- [GetProfileRequestMessage](#) ([User](#) requester)
Allocates a new.
- String [getType](#) ()
Returns the type of the message.
- void [process](#) ([IDataCom](#) IData, [InetAddress](#) senderAddress)
Sends a response to the user asking for the local user profile(user receiving this message) using `IDataCom` interface.

Package Attributes

- [User](#) [userRequester](#)

7.32.1 Detailed Description

This class implements the message which is sent when when a user wants to see the profile(including statistics) of an other user.

This class extends the abstract [Message](#) and implements the two abstract methods: `getType` and `process`

See also

[Message](#)

Author

COM Module

7.32.2 Constructor & Destructor Documentation

7.32.2.1 `lo23.battleship.online.network.messages.GetProfileRequestMessage.GetProfileRequestMessage (User requester)`

Allocates a new.

[GetProfileRequestMessage](#)

object.

Parameters

<i>requester</i>	: User : the user who requested to see the profile.
------------------	---

7.32.3 Member Function Documentation**7.32.3.1 String** `lo23.battleship.online.network.messages.GetProfileRequestMessage.getType ()`

Returns the type of the message.

Implementation of an abstract method.

Returns

type :

String

7.32.3.2 void `lo23.battleship.online.network.messages.GetProfileRequestMessage.process (IDataCom IData, InetAddress senderAddress)`

Sends a response to the user asking for the local user profile(user receiving this message) using IDataCom interface.

Parameters

<i>IData</i>	: IDataCom instance of IDataCom interface.
<i>senderAddress</i>	: InetAddress sender's IP address

7.32.4 Member Data Documentation**7.32.4.1 User** `lo23.battleship.online.network.messages.GetProfileRequestMessage.userRequester` [package]

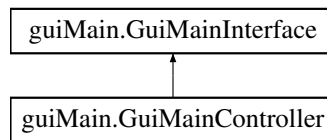
The documentation for this class was generated from the following file:

- `src/main/java/lo23/battleship/online/network/messages/GetProfileRequestMessage.java`

7.33 guiMain.GuiMainController Class Reference

This class implements the network controller.

Inheritance diagram for guiMain.GuiMainController:



Public Member Functions

- [IDataMain](#) [getldata](#) ()
Return the interface of data module that main use.
- void [addUser](#) (final [User](#) user)
Add a user to the menu list.
- void [removeUser](#) (final [User](#) user)
Remove a user to the menu list.
- void [addGame](#) (final [Game](#) createdGame)
Add a game to the menu list.
- void [removeGame](#) (final [Game](#) removedGame)
Remove a game to the menu list.
- void [sendStatistics](#) ([Profile](#) profil)
Set the statistic information for the profil window.
- void [transmitNewStatus](#) (final [Game](#) game)
Update a game status.
- void [setGameJoinResponse](#) (boolean isOk)
Updates according to the response to the request to join the game.
- void [startIHM](#) ()
Display the login window.
- void [openMenuWindow](#) ()
Display the menu window.
- void [openCreateGameWindow](#) ()
Open the window for creating a game.
- void [openSignupWindow](#) ()
Open the signup window.
- void [openConfigWindow](#) ()
Open the configuration window.
- void [openPlacementPhase](#) (final [Game](#) game)
Ask the IHM-Table module to open the placement phase window.
- void [openWaitingRoomWindow](#) ([Game](#) game)
Open the waiting room window.
- List< String > [getIps](#) ()
Return the list of ip address writen by the user.
- void [setIps](#) (List< String > list)
Set the list of ip address writen by the user.
- void [setldata](#) ([IDataMain](#) idata)
Set the data interface that would be use by the class.

- [GuiMainController](#) (Stage s)
Constructor for the class.
- void [askJoinGame](#) (final [Game](#) game)
Call data module to notify that the user want to join a game.
- void [openProfileWindow](#) ([User](#) user)
Open the profile window for a user.
- void [openChangeProfileWindow](#) ([User](#) user)
Open the profil window for the actual user.
- void [setPort](#) (int num_port)
Set the port that would be used to communicated between application.
- int [getPort](#) ()
Return the port that would be used to communicated between application.
- void [lookGame](#) ([Game](#) game)
Ask IHM-Table to display the observer phase of a game.

Package Attributes

- List< [User](#) > [playersList](#)
- List< [Game](#) > [gamesList](#)
- List< String > [ipsList](#) = new ArrayList<String>()
- int [port](#) = [Profile.DEFAULT_PORT](#)

Private Attributes

- Stage [stage](#)
- AnchorPane [rootLayout](#)
- [IDataMain](#) [idata](#)
- [menuController](#) [menuController](#)
- [SignupController](#) [signUpController](#)
- [IpConfigController](#) [ipConfigController](#)
- [LoginController](#) [loginController](#)
- [ProfilController](#) [profilController](#)
- [ChangeProfileController](#) [changeProfileController](#)
- [CreateGameController](#) [createGameController](#)
- [WaitingRoomController](#) [waitingRoomController](#)

7.33.1 Detailed Description

This class implements the network controller.

It contains references to all Main-Gui controllers and interactions with data module.

Author

IHM-Main module

7.33.2 Constructor & Destructor Documentation

7.33.2.1 guiMain.GuiMainController.GuiMainController (Stage s)

Constructor for the class.

Set the stage used by the application.

Parameters

<i>s</i>	: stage that would be used
----------	----------------------------

7.33.3 Member Function Documentation

7.33.3.1 void guiMain.GuiMainController.addGame (final Game *createdGame*)

Add a game to the menu list.

Parameters

<i>createdGame</i>	: the game that should be added
--------------------	---------------------------------

7.33.3.2 void guiMain.GuiMainController.addUser (final User *user*)

Add a user to the menu list.

Parameters

<i>user</i>	: the user that should be added
-------------	---------------------------------

7.33.3.3 void guiMain.GuiMainController.askJoinGame (final Game *game*)

Call data module to notify that the user want to join a game.

Parameters

<i>game</i>	: game that the user want to join
-------------	-----------------------------------

7.33.3.4 IDataMain guiMain.GuiMainController.getIdata ()

Return the interface of data module that main use.

Returns

IDataMain : the interface of data module

7.33.3.5 List<String> guiMain.GuiMainController.getIps ()

Return the list of ip address written by the user.

Returns

List<String> : ips list

7.33.3.6 int guiMain.GuiMainController.getPort ()

Return the port that would be used to communicated between application.

Returns

port used

7.33.3.7 void guiMain.GuiMainController.lookGame (Game game)

Ask IHM-Table to display the observer phase of a game.

Parameters

<i>game</i>	game that the user want to observe
-------------	------------------------------------

7.33.3.8 void guiMain.GuiMainController.openChangeProfileWindow (User user)

Open the profil window for the actual user.

The user can update his profile.

Parameters

<i>user</i>	: the actual user
-------------	-------------------

7.33.3.9 void guiMain.GuiMainController.openConfigWindow ()

Open the configuration window.

7.33.3.10 void guiMain.GuiMainController.openCreateGameWindow ()

Open the window for creating a game.

7.33.3.11 void guiMain.GuiMainController.openMenuWindow ()

Display the menu window.

Implements [guiMain.GuiMainInterface](#).

7.33.3.12 void guiMain.GuiMainController.openPlacementPhase (final Game game)

Ask the IHM-Table module to open the placement phase window.

7.33.3.13 void guiMain.GuiMainController.openProfileWindow (User user)

Open the profile window for a user.

Parameters

<i>user</i>	: user that the profil would be display
-------------	---

7.33.3.14 void guiMainGuiMainController.openSignupWindow ()

Open the signup window.

7.33.3.15 void guiMainGuiMainController.openWaitingRoomWindow (Game *game*)

Open the waiting room window.

Parameters

<i>game</i>	: the game that the user want to join
-------------	---------------------------------------

7.33.3.16 void guiMainGuiMainController.removeGame (final Game *removedGame*)

Remove a game to the menu list.

Parameters

<i>removedGame</i>	: the game that should be removed
--------------------	-----------------------------------

7.33.3.17 void guiMainGuiMainController.removeUser (final User *user*)

Remove a user to the menu list.

Parameters

<i>user</i>	: the user that should be removed
-------------	-----------------------------------

7.33.3.18 void guiMainGuiMainController.sendStatistics (Profile *profil*)

Set the statistic information for the profil window.

Parameters

<i>profil</i>	: the profil that should be displayed
---------------	---------------------------------------

Implements [guiMainGuiMainInterface](#).

7.33.3.19 void guiMain.GuiMainController.setGameJoinResponse (boolean *isOk*)

Updates according to the response to the request to join the game.

Parameters

<i>isOk</i>	: true : the request is accepted, false : the request is reject.
-------------	--

Implements [guiMain.GuiMainInterface](#).

7.33.3.20 void guiMain.GuiMainController.setldata (IDataMain *idata*)

Set the data interface that would be use by the class.

Parameters

<i>idata</i>	: data interface for main module
--------------	----------------------------------

7.33.3.21 void guiMain.GuiMainController.setIps (List< String > *list*)

Set the list of ip address written by the user.

Parameters

<i>list</i>	: ips list
-------------	------------

7.33.3.22 void guiMain.GuiMainController.setPort (int *num_port*)

Set the port that would be used to communicated between application.

Parameters

<i>num_port</i>	: port number
-----------------	---------------

7.33.3.23 void guiMain.GuiMainController.startIHM ()

Display the login window.

7.33.3.24 void guiMain.GuiMainController.transmitNewStatus (final Game *game*)

Update a game status.

Parameters

<i>game</i>	: the game that should be updated
-------------	-----------------------------------

7.33.4 Member Data Documentation

7.33.4.1 **ChangeProfileController** guiMainGuiMainController.changeProfileController [private]

7.33.4.2 **CreateGameController** guiMainGuiMainController.createGameController [private]

7.33.4.3 **List<Game>** guiMainGuiMainController.gamesList [package]

7.33.4.4 **IDataMain** guiMainGuiMainController.idata [private]

7.33.4.5 **IpConfigController** guiMainGuiMainController.ipConfigController [private]

7.33.4.6 **List<String>** guiMainGuiMainController.ipsList = new ArrayList<String>() [package]

7.33.4.7 **LoginController** guiMainGuiMainController.loginController [private]

7.33.4.8 **menuController** guiMainGuiMainController.menuController [private]

7.33.4.9 **List<User>** guiMainGuiMainController.playersList [package]

7.33.4.10 **int** guiMainGuiMainController.port = Profile.DEFAULT_PORT [package]

7.33.4.11 **ProfilController** guiMainGuiMainController.profilController [private]

7.33.4.12 **AnchorPane** guiMainGuiMainController.rootLayout [private]

7.33.4.13 **SignupController** guiMainGuiMainController.signUpController [private]

7.33.4.14 **Stage** guiMainGuiMainController.stage [private]

7.33.4.15 **WaitingRoomController** guiMainGuiMainController.waitingRoomController [private]

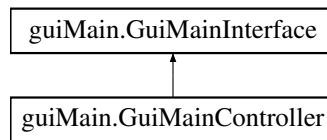
The documentation for this class was generated from the following file:

- src/main/java/guiMain/[GuiMainController.java](#)

7.34 guiMain.GuiMainInterface Interface Reference

GUI Main Interface.

Inheritance diagram for guiMain.GuiMainInterface:



Public Member Functions

- void [addUser](#) ([User](#) user)
Adds the user passed as a parameter to the list of users.
- void [removeUser](#) ([User](#) user)
Remove the user passed as a parameter to the list of users.
- void [addGame](#) ([Game](#) createdGame)
Adds the game passed as a parameter in the list of games displayed.
- void [sendStatistics](#) ([Profile](#) profil)
Displays the statistics of a player.
- void [transmitNewStatus](#) ([Game](#) game)
Updates the status of the game.
- void [setGameJoinResponse](#) (boolean isOk)
Updates according to the response to the request to join the game.
- void [openPlacementPhase](#) ([Game](#) game)
- void [removeGame](#) ([Game](#) game)
Remove the game passed as a parameter to the list of users.
- void [openMenuWindow](#) ()
To open the menu window.

7.34.1 Detailed Description

GUI Main Interface.

7.34.2 Member Function Documentation

7.34.2.1 void guiMain.GuiMainInterface.addGame ([Game](#) createdGame)

Adds the game passed as a parameter in the list of games displayed.

Parameters

<i>createdGame</i>	: game to be added to the list of games.
--------------------	--

7.34.2.2 void guiMain.GuiMainInterface.addUser (User *user*)

Adds the user passed as a parameter to the list of users.

Parameters

<i>user</i>	: user to add to the list.
-------------	----------------------------

7.34.2.3 void guiMain.GuiMainInterface.openMenuWindow ()

To open the menu window.

Implemented in [guiMain.GuiMainController](#).

7.34.2.4 void guiMain.GuiMainInterface.openPlacementPhase (Game *game*)

7.34.2.5 void guiMain.GuiMainInterface.removeGame (Game *game*)

Remove the game passed as a parameter to the list of users.

Parameters

<i>game</i>	: game to remove to the list.
-------------	-------------------------------

7.34.2.6 void guiMain.GuiMainInterface.removeUser (User *user*)

Remove the user passed as a parameter to the list of users.

Parameters

<i>user</i>	: user to remove to the list.
-------------	-------------------------------

7.34.2.7 void guiMain.GuiMainInterface.sendStatistics (Profile *profil*)

Displays the statistics of a player.

Parameters

<i>profil</i>	: profile of the player.
---------------	--------------------------

Implemented in [guiMain.GuiMainController](#).

7.34.2.8 void guiMain.GuiMainInterface.setGameJoinResponse (boolean *isOk*)

Updates according to the response to the request to join the game.

Parameters

<i>isOk</i>	: true : the request is accepted, false : the request is reject.
-------------	--

Implemented in [guiMain.GuiMainController](#).

7.34.2.9 void guiMain.GuiMainInterface.transmitNewStatus (Game *game*)

Updates the status of the game.

Parameters

<i>game</i>	: Game is updated status.
-------------	---------------------------

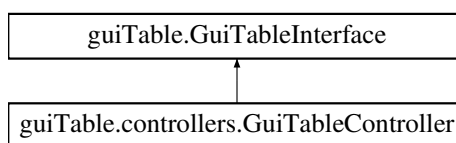
The documentation for this interface was generated from the following file:

- [src/main/java/guiMain/GuiMainInterface.java](#)

7.35 guiTable.controllers.GuiTableController Class Reference

[GuiTableController](#).

Inheritance diagram for guiTable.controllers.GuiTableController:



Public Member Functions

- void [displayPlacementPhase](#) (Stage currentStage, Boolean [classic](#), int placementTime) throws Exception
this function call an other fxml context and refresh page
- void [opponentReady](#) (final Boolean myTurn, long timePerShot)
- void [displayObserverPhase](#) (Stage currentStage, [Game](#) game)
Displays view where the user can watch a game.
- void [updateSpectatorGame](#) ([Game](#) game)
- void [displayObserverShot](#) (final [Shot](#) shot, int player)
- void [displayObserverPhaseVictory](#) (int winner)
Displays the result of the player 1's shot on the player 2's board.
- void [displayVictory](#) ()

- *Displays a victory message, and allow player to return to IHM Main.*
- void [displayDefeat](#) ()
- *Displays a defeat message, and allow player to return to IHM Main.*
- void [addChatMessage](#) ([ChatMessage](#) message)
- *Displays a new incoming message on chat window.*
- void [displayMyShotResult](#) (final [Shot](#) myShotResult, final [Boat](#) boat)
- void [displayOpponentShot](#) (final [Shot](#) opponentShot, final [Boat](#) boat)
- void [setDataController](#) ([CDataTable](#) data)
- void [validateBoats](#) (List< [Boat](#) > boats)
- void [validateShot](#) ([Position](#) pos)
- Boolean [exitGame](#) ()
- [CDataTable](#) [getDataController](#) ()
- void [displayRageQuit](#) ()
- void [sunkPlayerBoat](#) (int i, [Boat](#) boat)

Static Public Member Functions

- static [GuiTableController](#) [getInstance](#) ()
- *Entry point for a unique instance of singleton [GuiTableController](#);*

Private Member Functions

- [GuiTableController](#) ()
- *Private constructor for [GuiTableController](#).*

Private Attributes

- AnchorPane [rootLayout](#)
- Stage [mainStage](#)
- Boolean [classic](#)
- [GamePhaseController](#) [gamePhaseController](#)
- [ObserverPhaseController](#) [observerPhaseController](#)
- [ObservationPhase](#) [observationControlleur](#)
- [CDataTable](#) [dataController](#)
- [ChatController](#) [chatController](#)
- [PlacementPhaseController](#) [placementPhaseController](#)
- final String [CHAT_FXML_URL](#) = "/fxml/IhmTable/chat.fxml"
- final String [CLASSIC_PLACEMENT_URL](#) = "/fxml/IhmTable/ClassicPlacementPhase.fxml"
- final String [BELGE_PLACEMENT_URL](#) = "/fxml/IhmTable/BelgianPlacementPhase.fxml"
- final String [GAME_PHASE_URL](#) = "/fxml/IhmTable/GamePhase.fxml"
- final String [OBSERVER_PHASE_URL](#) = "/fxml/IhmTable/ObserverPhase.fxml"
- final String [OBSERVATION_PHASE_URL](#) = "/fxml/IhmTable/observationPhase.fxml"
- final String [TITLE](#) = "Battleship-Online"
- List< [Boat](#) > [boats](#) = null

Static Private Attributes

- static [GuiTableController](#) [INSTANCE](#) = null

7.35.1 Detailed Description

[GuiTableController](#).

7.35.2 Constructor & Destructor Documentation

7.35.2.1 `guiTable.controllers.GuiTableController.GuiTableController () [private]`

Private constructor for [GuiTableController](#).

7.35.3 Member Function Documentation

7.35.3.1 `void guiTable.controllers.GuiTableController.addChatMessage (ChatMessage message)`

Displays a new incoming message on chat window.

Parameters

<i>message</i>	: New incoming message to display.
----------------	------------------------------------

Implements [guiTable.GuiTableInterface](#).

7.35.3.2 `void guiTable.controllers.GuiTableController.displayDefeat ()`

Displays a defeat message, and allow player to return to IHM Main.

Implements [guiTable.GuiTableInterface](#).

7.35.3.3 `void guiTable.controllers.GuiTableController.displayMyShotResult (final Shot myShotResult, final Boat boat)`

7.35.3.4 `void guiTable.controllers.GuiTableController.displayObserverPhase (Stage currentStage, Game game)`

Displays view where the user can watch a game.

Parameters

<i>currentStage</i>	: The stage used by the main window
<i>game</i>	

Implements [guiTable.GuiTableInterface](#).

7.35.3.5 `void guiTable.controllers.GuiTableController.displayObserverPhaseVictory (int winner)`

Displays the result of the player 1's shot on the player 2's board.

Parameters

<i>shot</i>	: The shot sent by the player 1.
<i>boat</i>	: If filled, indicates this 'boat' is sunk. Displays the result of the player 2's shot on the player 1's board.
<i>shot</i>	: The shot sent by the player 2.
<i>boat</i>	: If filled, indicates this 'boat' is sunk. Displays the victory of one of the players in the observer phase
<i>winner</i>	the position of the winner

Implements [guiTable.GuiTableInterface](#).

7.35.3.6 void guiTable.controllers.GuiTableController.displayObserverShot (final Shot *shot*, int *player*)

7.35.3.7 void guiTable.controllers.GuiTableController.displayOpponentShot (final Shot *opponentShot*, final Boat *boat*)

7.35.3.8 void guiTable.controllers.GuiTableController.displayPlacementPhase (Stage *currentStage*, Boolean *classic*, int *placementTime*) throws Exception

this function call an other fxml context and refresh page

Parameters

<i>currentStage</i>	
<i>placementTime</i>	

Exceptions

<i>Exception</i>	
------------------	--

Implements [guiTable.GuiTableInterface](#).

7.35.3.9 void guiTable.controllers.GuiTableController.displayRageQuit ()

Implements [guiTable.GuiTableInterface](#).

7.35.3.10 void guiTable.controllers.GuiTableController.displayVictory ()

Displays a victory message, and allow player to return to IHM Main.

Implements [guiTable.GuiTableInterface](#).

7.35.3.11 `Boolean guiTable.controllers.GuiTableController.exitGame ()`

7.35.3.12 `CDataTable guiTable.controllers.GuiTableController.getDataController ()`

7.35.3.13 `static GuiTableController guiTable.controllers.GuiTableController.getInstance ()` [static]

Entry point for a unique instance of singleton [GuiTableController](#);

Returns

[GuiTableController](#) : the singleton [GuiTableController](#).

7.35.3.14 `void guiTable.controllers.GuiTableController.opponentReady (final Boolean myTurn, long timePerShot)`

7.35.3.15 `void guiTable.controllers.GuiTableController.setDataController (CDataTable data)`

Implements [guiTable.GuiTableInterface](#).

7.35.3.16 `void guiTable.controllers.GuiTableController.sunkPlayerBoat (int i, Boat boat)`

Implements [guiTable.GuiTableInterface](#).

7.35.3.17 `void guiTable.controllers.GuiTableController.updateSpectatorGame (Game game)`

Implements [guiTable.GuiTableInterface](#).

7.35.3.18 `void guiTable.controllers.GuiTableController.validateBoats (List< Boat > boats)`

7.35.3.19 `void guiTable.controllers.GuiTableController.validateShot (Position pos)`

7.35.4 Member Data Documentation

7.35.4.1 `final String guiTable.controllers.GuiTableController.BELGE_PLACEMENT_URL = "/fxml/lhmTable/BelgianPlacement↵Phase.fxml" [private]`

7.35.4.2 `List<Boat> guiTable.controllers.GuiTableController.boats = null [private]`

7.35.4.3 `final String guiTable.controllers.GuiTableController.CHAT_FXML_URL = "/fxml/lhmTable/chat.fxml" [private]`

7.35.4.4 `ChatController guiTable.controllers.GuiTableController.chatController [private]`

7.35.4.5 `Boolean guiTable.controllers.GuiTableController.classic [private]`

- 7.35.4.6 `final String guiTable.controllers.GuiTableController.CLASSIC_PLACEMENT_URL =
"/fxml/lhmTable/ClassicPlacementPhase.fxml" [private]`
- 7.35.4.7 `CDataTable guiTable.controllers.GuiTableController.dataController [private]`
- 7.35.4.8 `final String guiTable.controllers.GuiTableController.GAME_PHASE_URL = "/fxml/lhmTable/GamePhase.fxml"
[private]`
- 7.35.4.9 `GamePhaseController guiTable.controllers.GuiTableController.gamePhaseController [private]`
- 7.35.4.10 `GuiTableController guiTable.controllers.GuiTableController.INSTANCE = null [static], [private]`
- 7.35.4.11 `Stage guiTable.controllers.GuiTableController.mainStage [private]`
- 7.35.4.12 `final String guiTable.controllers.GuiTableController.OBSERVATION_PHASE_URL =
"/fxml/lhmTable/observationPhase.fxml" [private]`
- 7.35.4.13 `ObservationPhase guiTable.controllers.GuiTableController.observationController [private]`
- 7.35.4.14 `final String guiTable.controllers.GuiTableController.OBSERVER_PHASE_URL = "/fxml/lhmTable/ObserverPhase.fxml"
[private]`
- 7.35.4.15 `ObserverPhaseController guiTable.controllers.GuiTableController.observerPhaseController [private]`
- 7.35.4.16 `PlacementPhaseController guiTable.controllers.GuiTableController.placementPhaseController [private]`
- 7.35.4.17 `AnchorPane guiTable.controllers.GuiTableController.rootLayout [private]`
- 7.35.4.18 `final String guiTable.controllers.GuiTableController.TITLE = "Battleship-Online" [private]`

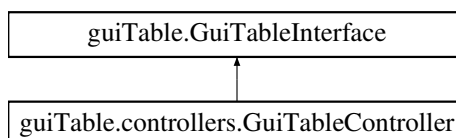
The documentation for this class was generated from the following file:

- `src/main/java/guiTable/controllers/GuiTableController.java`

7.36 guiTable.GuiTableInterface Interface Reference

Interface for the lhm Table team.

Inheritance diagram for guiTable.GuiTableInterface:



Public Member Functions

- void [displayPlacementPhase](#) (Stage currentStage, Boolean classic, int timePerShot) throws Exception
Displays the window where the player will place their ships.
- void [opponentReady](#) (Boolean myTurn, long time)
Notifies the IHM Table that the opponent is ready so that it can adapt its views.
- void [displayObserverPhase](#) (Stage currentStage, [Game](#) game)
Displays view where the user can watch a game.
- void [displayObserverShot](#) ([Shot](#) shot, int player)
Displays the result of the shot send by player number player.
- void [displayObserverPhaseVictory](#) (int winner)
Displays the result of the player 1's shot on the player 2's board.
- void [displayVictory](#) ()
Displays a victory message, and allow player to return to IHM Main.
- void [displayDefeat](#) ()
Displays a defeat message, and allow player to return to IHM Main.
- void [displayOpponentShot](#) ([Shot](#) opponentShot, [Boat](#) boat)
Displays the result of the opponent shot on the board.
- void [displayMyShotResult](#) ([Shot](#) myShotResult, [Boat](#) boat)
Displays the result of the user shot on the board.
- void [addChatMessage](#) ([ChatMessage](#) message)
Displays a new incoming message on chat window.
- void [setDataController](#) ([CDataTable](#) d)
- void [displayRageQuit](#) ()
- void [updateSpectatorGame](#) ([Game](#) g)
- void [sunkPlayerBoat](#) (int i, [Boat](#) boat)

7.36.1 Detailed Description

Interface for the Ihm Table team.

7.36.2 Member Function Documentation

7.36.2.1 void [guiTable.GuiTableInterface.addChatMessage](#) ([ChatMessage](#) *message*)

Displays a new incoming message on chat window.

Parameters

<i>message</i>	: New incoming message to display.
----------------	------------------------------------

Implemented in [guiTable.controllers.GuiTableController](#).

7.36.2.2 void [guiTable.GuiTableInterface.displayDefeat](#) ()

Displays a defeat message, and allow player to return to IHM Main.

Implemented in [guiTable.controllers.GuiTableController](#).

7.36.2.3 void guiTable.GuiTableInterface.displayMyShotResult (Shot *myShotResult*, Boat *boat*)

Displays the result of the user shot on the board.

Parameters

<i>myShotResult</i>	: The result of the shot sent by the user.
<i>boat</i>	: If filled, indicates this 'boat' is sunk.

7.36.2.4 void guiTable.GuiTableInterface.displayObserverPhase (Stage *currentStage*, Game *game*)

Displays view where the user can watch a game.

Parameters

<i>currentStage</i>	: The stage used by the main window
<i>game</i>	

Implemented in [guiTable.controllers.GuiTableController](#).

7.36.2.5 void guiTable.GuiTableInterface.displayObserverPhaseVictory (int *winner*)

Displays the result of the player 1's shot on the player 2's board.

Parameters

<i>shot</i>	: The shot sent by the player 1.
<i>boat</i>	: If filled, indicates this 'boat' is sunk. Displays the result of the player 2's shot on the player 1's board.
<i>shot</i>	: The shot sent by the player 2.
<i>boat</i>	: If filled, indicates this 'boat' is sunk. Displays the victory of one of the players in the observer phase
<i>winner</i>	the position of the winner

Implemented in [guiTable.controllers.GuiTableController](#).

7.36.2.6 void guiTable.GuiTableInterface.displayObserverShot (Shot *shot*, int *player*)

Displays the result of the shot send by player number player.

Parameters

<i>shot</i>	: The shot sent by the player 1.
<i>player</i>	: the player who send the shot

7.36.2.7 void guiTable.GuiTableInterface.displayOpponentShot (Shot *opponentShot*, Boat *boat*)

Displays the result of the opponent shot on the board.

Parameters

<i>opponentShot</i>	: The shot sent by the opponent.
<i>boat</i>	: If filled, indicates this 'boat' is sunk.

7.36.2.8 void guiTable.GuiTableInterface.displayPlacementPhase (Stage *currentStage*, Boolean *classic*, int *timePerShot*) throws Exception

Displays the window where the player will place their ships.

Parameters

<i>currentStage</i>	current stage to be replaced
<i>classic</i>	specify if it is a classic or belgian game
<i>timePerShot</i>	: Time limitation for placement phase, if null, no time limitation

Exceptions

<i>Exception</i>	
------------------	--

Implemented in [guiTable.controllers.GuiTableController](#).

7.36.2.9 void guiTable.GuiTableInterface.displayRageQuit ()

Implemented in [guiTable.controllers.GuiTableController](#).

7.36.2.10 void guiTable.GuiTableInterface.displayVictory ()

Displays a victory message, and allow player to return to IHM Main.

Implemented in [guiTable.controllers.GuiTableController](#).

7.36.2.11 void guiTable.GuiTableInterface.opponentReady (Boolean *myTurn*, long *time*)

Notifies the IHM Table that the opponent is ready so that it can adapt its views.

Without round time limitation This method exist to avoid BC compatibilities

Parameters

<i>myTurn</i>	specify if it is the local player turn
<i>time</i>	the time

7.36.2.12 void `guiTable.GuiTableInterface.setDataController (CDataTable d)`

Implemented in [guiTable.controllers.GuiTableController](#).

7.36.2.13 void `guiTable.GuiTableInterface.sunkPlayerBoat (int i, Boat boat)`

Implemented in [guiTable.controllers.GuiTableController](#).

7.36.2.14 void `guiTable.GuiTableInterface.updateSpectatorGame (Game g)`

Implemented in [guiTable.controllers.GuiTableController](#).

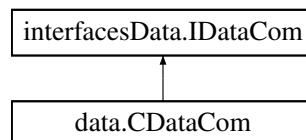
The documentation for this interface was generated from the following file:

- `src/main/java/guiTable/GuiTableInterface.java`

7.37 interfacesData.IDataCom Interface Reference

Data's interface for Com.

Inheritance diagram for interfacesData.IDataCom:



Public Member Functions

- [Game](#) `getCreatedGame ()`
Returns the current Game.
- void `setGameJoinResponse` (Boolean ok, [Player](#) player1, [Player](#) player2)
The distant user has accepted or not the request to join the game and the method updateGameData will be used to update the game data.
- void `setGameJoinResponse` (Boolean no)
The distance user has refused the request to join the game.
- void `addUserToUserList` ([User](#) u)
After an user has connected, this user will be added to the list of user.
- void `sendStatistics` ([Profile](#) p)
Sends the profile of a distant user to the local user so that the local user can see the profile of this distant user.
- void `notifToJoinGame` ([Profile](#) sender, [Game](#) g)
Add the player to the game if it is available.
- void `addNewGameList` ([Game](#) g)
Adds the game given as a parameter to the list of games.
- void `removeGameFromList` ([Game](#) g)
Removes the game given as a parameter from the list of games.

- void `receiveMessage` (`ChatMessage` message)
Takes the chat message given as a parameter in order to transmit it to IHM-Table.
- void `receiveReady` ()
Indicates that a player is ready to play (all his boats are placed on his table) so that the shots phase can be displayed.
- void `coordinates` (`Shot` s, `Boat` b)
Takes a Shot and in option a Boat to transmit it to IHM-Table.
- void `coordinates` (`Shot` s)
Takes a Shot to transmit it to IHM-Table.
- `Profile` `getUserProfile` ()
Returns the local user's profile, containing his statistics.
- void `changeStatusGame` (`Game` g)
Takes a game given as a parameter and updates his status.
- `User` `getLocalUser` ()
Accessor for the local User.
- void `removeUser` (`User` u)
To remove a User.
- void `setLocalGame` (`Game` g)
Set the local Game with the game given as a parameter.
- void `removeGame` (`Game` g)
Remove a Game from local list.
- void `notifiedGameWon` ()
Notification that you won, update stats and display win.
- void `notifyToSpecGame` (`User` spec)
Notify that a new spectator has joined the game.
- void `newRequestSpectator` (`User` u)
A new spectator want to join the game, he need to get the informations of the game.
- void `joinGameSpectator` (`Game` g)
The spectator receives the informations of the game that he wants to join.
- void `notifyQuitSpectator` (`User` spec)
Notif everyone when a spectator leaves.
- `Player` `getOtherPlayer` ()
Accessor for the game's other player.
- void `updateAttendedGame` (`Player` p, `Shot` s, `Boat` b)
Update attended in a game.
- void `notifyAttendedGameWon` (`Player` p)
Notify when a player won a game.
- int `getPlayerPosition` (`Player` p, `Game` g)
Accessor for a player's position.

7.37.1 Detailed Description

Data's interface for Com.

7.37.2 Member Function Documentation

7.37.2.1 void `interfacesData.IDataCom.addNewGameList` (`Game` g)

Adds the game given as a parameter to the list of games.

Parameters

<i>g</i>	: The new game
----------	----------------

Implemented in [data.CDataCom](#).

7.37.2.2 void interfacesData.IDataCom.addUserToUserList (User *u*)

After an user has connected, this user will be added to the list of user.

Parameters

<i>u</i>	: The new user
----------	----------------

Implemented in [data.CDataCom](#).

7.37.2.3 void interfacesData.IDataCom.changeStatusGame (Game *g*)

Takes a game given as a parameter and updates his status.

Parameters

<i>g</i>	: The game which status has to be modified
----------	--

Implemented in [data.CDataCom](#).

7.37.2.4 void interfacesData.IDataCom.coordinates (Shot *s*, Boat *b*)

Takes a Shot and in option a Boat to transmit it to IHM-Table.

Parameters

<i>s</i>	: The position played by the user
<i>b</i>	: In option, the boat that was sunk

Implemented in [data.CDataCom](#).

7.37.2.5 void interfacesData.IDataCom.coordinates (Shot *s*)

Takes a Shot to transmit it to IHM-Table.

Parameters

<i>s</i>	: The position played by the user
----------	-----------------------------------

Implemented in [data.CDataCom](#).

7.37.2.6 Game interfaces `Data.IDataCom.getCreatedGame ()`

Returns the current Game.

Returns

the current Game

Implemented in [data.CDataCom](#).

7.37.2.7 User interfaces `Data.IDataCom.getLocalUser ()`

Accessor for the local User.

Returns

public void [removeUser\(User u\)](#)

Implemented in [data.CDataCom](#).

7.37.2.8 Player interfaces `Data.IDataCom.getOtherPlayer ()`

Accessor for the game's other player.

Returns

the other player

Implemented in [data.CDataCom](#).

7.37.2.9 `int` interfaces `Data.IDataCom.getPlayerPosition (Player p, Game g)`

Accessor for a player's position.

Parameters

<i>p</i>	player
<i>g</i>	game

Returns

the player's position

Implemented in [data.CDataCom](#).

7.37.2.10 Profile interfacesData.IDataCom.getUserProfile ()

Returns the local user's profile, containing his statistics.

Returns

the local user's profile

Implemented in [data.CDataCom](#).

7.37.2.11 void interfacesData.IDataCom.joinGameSpectator (Game *g*)

The spectator receives the informations of the game that he wants to join.

Parameters

<i>g</i>	The joined game
----------	-----------------

Implemented in [data.CDataCom](#).

7.37.2.12 void interfacesData.IDataCom.newRequestSpectator (User *u*)

A new spectator want to join the game, he need to get the informations of the game.

Parameters

<i>u</i>	The spectator who want to come
----------	--------------------------------

Implemented in [data.CDataCom](#).

7.37.2.13 void interfacesData.IDataCom.notifiedGameWon ()

Notification that you won, update stats and display win.

Implemented in [data.CDataCom](#).

7.37.2.14 void interfacesData.IDataCom.notifToJoinGame (Profile *sender*, Game *g*)

Add the player to the game if it is available.

Parameters

<i>sender</i>	: The player who sends this request
<i>g</i>	: The game that the player wants to join

Implemented in [data.CDataCom](#).

7.37.2.15 void interfacesData.IDataCom.notifyAttendedGameWon (Player *p*)

Notify when a player won a game.

Parameters

<i>p</i>	the player
----------	------------

Implemented in [data.CDataCom](#).

7.37.2.16 void interfacesData.IDataCom.notifyQuitSpectator (User *spec*)

Notif everyone when a spectator leaves.

Parameters

<i>spec</i>	The spectator who leaves
-------------	--------------------------

Implemented in [data.CDataCom](#).

7.37.2.17 void interfacesData.IDataCom.notifyToSpecGame (User *spec*)

Notify that a new spectator has joined the game.

Parameters

<i>spec</i>	New spectator
-------------	---------------

Implemented in [data.CDataCom](#).

7.37.2.18 void interfacesData.IDataCom.receiveMessage (ChatMessage *message*)

Takes the chat message given as a parameter in order to transmit it to IHM-Table.

Parameters

<i>message</i>	: The chat message to transmit
----------------	--------------------------------

Implemented in [data.CDataCom](#).

7.37.2.19 void interfacesData.IDataCom.receiveReady ()

Indicates that a player is ready to play (all his boats are placed on his table) so that the shots phase can be displayed.

Implemented in [data.CDataCom](#).

7.37.2.20 void interfacesData.IDataCom.removeGame (Game *g*)

Remove a Game from local list.

Parameters

<i>g</i>	: Game to remove
----------	------------------

Implemented in [data.CDataCom](#).

7.37.2.21 void interfacesData.IDataCom.removeGameFromList (Game *g*)

Removes the game given as a parameter from the list of games.

Parameters

<i>g</i>	: game to remove
----------	------------------

Implemented in [data.CDataCom](#).

7.37.2.22 void interfacesData.IDataCom.removeUser (User *u*)

To remove a User.

Parameters

<i>u</i>	User to remove
----------	----------------

Implemented in [data.CDataCom](#).

7.37.2.23 void interfacesData.IDataCom.sendStatistics (Profile *p*)

Sends the profile of a distant user to the local user so that the local user can see the profile of this distant user.

Parameters

<i>p</i>	The profile to be sent
----------	------------------------

Implemented in [data.CDataCom](#).

7.37.2.24 void interfacesData.IDataCom.setGameJoinResponse (Boolean *ok*, Player *player1*, Player *player2*)

The distant user has accepted or not the request to join the game and the method `updateGameData` will be used to update the game data.

Parameters

<i>ok</i>	: Acceptance of the request to join the game
<i>player1</i>	: Creator of the game
<i>player2</i>	: The player who joins the game

Implemented in [data.CDataCom](#).

7.37.2.25 void interfacesData.IDataCom.setGameJoinResponse (Boolean *no*)

The distance user has refused the request to join the game.

Parameters

<i>no</i>	: Refuse of the request to join the game
-----------	--

Implemented in [data.CDataCom](#).

7.37.2.26 void interfacesData.IDataCom.setLocalGame (Game *g*)

Set the local Game with the game given as a parameter.

Parameters

<i>g</i>	: new value for the local Game
----------	--------------------------------

Implemented in [data.CDataCom](#).

7.37.2.27 void interfacesData.IDataCom.updateAttendedGame (Player *p*, Shot *s*, Boat *b*)

Update attended in a game.

Parameters

<i>p</i>	a player
<i>s</i>	a shot
<i>b</i>	a boat

Implemented in [data.CDataCom](#).

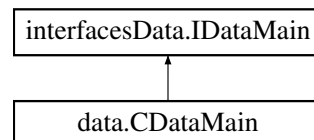
The documentation for this interface was generated from the following file:

- [src/main/java/interfacesData/IDataCom.java](#)

7.38 interfacesData.IDataMain Interface Reference

Data's interface for Main.

Inheritance diagram for interfacesData.IDataMain:



Public Member Functions

- void [editProfile](#) (String username, String password, String avatar, String lastName, String firstName, Date borthDate)
Modify the local profile.
- void [createAccount](#) (String login, String username, HashSet ips, int port, String password, List< [ContactGroup](#) > contactList, String avatar, String lastname, String firstname, Date birthDate)
Create a local account.
- void [getProfile](#) ([User](#) u)
Returns the profile of an user.
- void [notifGameChosen](#) ([Game](#) g)
Notifies the away application that an user wants to join a game.
- void [askDisconnection](#) ()
Notifies away applications that the local user disconnects and erases his session.
- Boolean [connection](#) (String login, String password) throws UnknownHostException
Loads the saved data of the user and researches players.
- [Game](#) [newGame](#) (Boolean newClassicType, String newName, Boolean newHumanOpponent, Integer newTimePerShot, Integer newTimeToPlaceBoats, Boolean newSpectator, Boolean newSpectatorChat)
Add a new game to the list of games.
- void [removeGame](#) ([Game](#) g)
Remove a Game from local list.
- List< [Game](#) > [getGames](#) ()
- [Profile](#) [getLocalProfile](#) ()
- void [setLocalGame](#) ([Game](#) g)
Set the local game.
- void [setListIps](#) (HashSet Ips)
Set the list of IP adress.
- void [gameToSpec](#) ([Game](#) g)
Add a spectator in the game.
- void [setPort](#) (int p)
Set the port.
- List< [User](#) > [getListUsers](#) ()
- void [clear](#) ()

7.38.1 Detailed Description

Data's interface for Main.

7.38.2 Member Function Documentation

7.38.2.1 void interfacesData.IDataMain.askDisconnection ()

Notifies away applications that the local user disconnects and erases his session.

Implemented in [data.CDataMain](#).

7.38.2.2 void interfacesData.IDataMain.clear ()

Implemented in [data.CDataMain](#).

7.38.2.3 Boolean interfacesData.IDataMain.connection (String *login*, String *password*) throws UnknownHostException

Loads the saved data of the user and researches players.

Parameters

<i>login</i>	the login
<i>password</i>	the password

Returns

a boolean

Exceptions

<i>java.net.UnknownHostException</i>	
--------------------------------------	--

Implemented in [data.CDataMain](#).

7.38.2.4 void interfacesData.IDataMain.createAccount (String *login*, String *username*, HashSet *ips*, int *port*, String *password*, List< **ContactGroup** > *contactList*, String *avatar*, String *lastname*, String *firstname*, Date *birthDate*)

Create a local account.

Parameters

<i>login</i>	: the user's login
<i>username</i>	: the user's username
<i>ips</i>	: list of the IP adresses known by the user

Parameters

<i>port</i>	: the port
<i>password</i>	: the user's password
<i>contactList</i>	: list of the user's contacts
<i>avatar</i>	: a path to the user's avatar
<i>lastname</i>	: the user's lastname
<i>firstname</i>	: the user's firstname
<i>birthDate</i>	: the user's birthdate

Implemented in [data.CDataMain](#).

7.38.2.5 `void interfacesData.IDataMain.editProfile (String username, String password, String avatar, String lastName, String firstName, Date borthDate)`

Modify the local profile.

Parameters

<i>username</i>	: new username
<i>password</i>	: new password
<i>avatar</i>	: new avatar
<i>lastName</i>	: new lastName
<i>firstName</i>	: new firstName
<i>borthDate</i>	: new birthDate

Implemented in [data.CDataMain](#).

7.38.2.6 `void interfacesData.IDataMain.gameToSpec (Game g)`

Add a spectator in the game.

Parameters

<i>g</i>	: game that the spectator wants to join
----------	---

Implemented in [data.CDataMain](#).

7.38.2.7 `List<Game> interfacesData.IDataMain.getGames ()`

Returns

the local list of games

Implemented in [data.CDataMain](#).

7.38.2.8 `List<User> interfacesData.IDataMain.getListUsers ()`**Returns**

the local list of users

Implemented in [data.CDataMain](#).

7.38.2.9 `Profile interfacesData.IDataMain.getLocalProfile ()`**Returns**

the local profile

Implemented in [data.CDataMain](#).

7.38.2.10 `void interfacesData.IDataMain.getProfile (User u)`

Returns the profile of an user.

Parameters

<i>u</i>	the user
----------	----------

Implemented in [data.CDataMain](#).

7.38.2.11 `Game interfacesData.IDataMain.newGame (Boolean newClassicType, String newName, Boolean newHumanOpponent, Integer newTimePerShot, Integer newTimeToPlaceBoats, Boolean newSpectator, Boolean newSpectatorChat)`

Add a new game to the list of games.

Parameters

<i>newClassicType</i>	the type
<i>newName</i>	the name
<i>newHumanOpponent</i>	if is against a human or not
<i>newTimePerShot</i>	the time per shot
<i>newTimeToPlaceBoats</i>	the time to place boats
<i>newSpectator</i>	if spectators are allowed
<i>newSpectatorChat</i>	if chat is allowed

Returns

the new game

Implemented in [data.CDataMain](#).

7.38.2.12 void interfacesData.IDataMain.notifGameChosen (Game *g*)

Notifies the away application that an user wants to join a game.

Parameters

<i>g</i>	: Game the user wants to join
----------	-------------------------------

Implemented in [data.CDataMain](#).

7.38.2.13 void interfacesData.IDataMain.removeGame (Game *g*)

Remove a Game from local list.

Parameters

<i>g</i>	: Game to remove
----------	------------------

Implemented in [data.CDataMain](#).

7.38.2.14 void interfacesData.IDataMain.setListIps (HashSet *lps*)

Set the list of IP adress.

Parameters

<i>lps</i>	: list of IP
------------	--------------

Implemented in [data.CDataMain](#).

7.38.2.15 void interfacesData.IDataMain.setLocalGame (Game *g*)

Set the local game.

Parameters

<i>g</i>	: game to set
----------	---------------

Implemented in [data.CDataMain](#).

7.38.2.16 void interfacesData.IDataMain.setPort (int *p*)

Set the port.

Parameters

<i>p</i>	: the port to set
----------	-------------------

Implemented in [data.CDataMain](#).

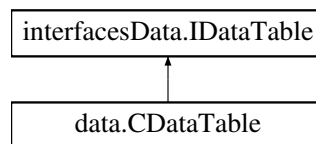
The documentation for this interface was generated from the following file:

- [src/main/java/interfacesData/IDataMain.java](#)

7.39 interfacesData.IDataTable Interface Reference

Data's interface for IHM-Table.

Inheritance diagram for interfacesData.IDataTable:



Public Member Functions

- Boolean [exit](#) ()
Function to exit.
- void [textMessage](#) (String message)
Add the message to the chat in the current game.
- void [coordinate](#) ([Position](#) pos)
Point out the position of shot.
- void [coordinateShips](#) (List< [Boat](#) > listBoat)
Point out the the boats that players place at the beginning of game.
- void [timerOver](#) ()
- void [changeStatusGameStarted](#) ()
Transmit to every user that the game is beign played, allow them to spectate.
- [Game](#) [getObserverGame](#) ()
Accessor for the observed game.
- void [gameEnded](#) ()
To end a game.
- [Game](#) [getLocalGame](#) ()

7.39.1 Detailed Description

Data's interface for IHM-Table.

7.39.2 Member Function Documentation

7.39.2.1 void interfacesData.IDataTable.changeStatusGameStarted ()

Transmit to every user that the game is beign played, allow them to spectate.

Implemented in [data.CDataTable](#).

7.39.2.2 void interfacesData.IDataTable.coordinate (Position *pos*)

Point out the position of shot.

Parameters

<i>pos</i>	: The position of shot.
------------	-------------------------

Implemented in [data.CDataTable](#).

7.39.2.3 void interfacesData.IDataTable.coordinateShips (List< Boat > *listBoat*)

Point out the the boats that players place at the beginning of game.

Parameters

<i>listBoat</i>	: List of boats.
-----------------	------------------

Implemented in [data.CDataTable](#).

7.39.2.4 Boolean interfacesData.IDataTable.exit ()

Function to exit.

Returns

: 1 if the game was successfully closed and return 0 if not.

Implemented in [data.CDataTable](#).

7.39.2.5 void interfacesData.IDataTable.gameEnded ()

To end a game.

Implemented in [data.CDataTable](#).

7.39.2.6 Game interfacesData.IDataTable.getLocalGame ()

Returns

the local game

Implemented in [data.CDataTable](#).

7.39.2.7 Game interfacesData.IDataTable.getObserverGame ()

Accessor for the observed game.

Returns

the game observed

Implemented in [data.CDataTable](#).

7.39.2.8 void interfacesData.IDataTable.textMessage (String *message*)

Add the message to the chat in the current game.

Parameters

<i>message</i>	: The main part of message that the player wants to send.
----------------	---

Implemented in [data.CDataTable](#).

7.39.2.9 void interfacesData.IDataTable.timerOver ()

Implemented in [data.CDataTable](#).

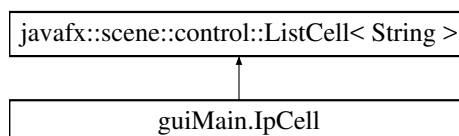
The documentation for this interface was generated from the following file:

- [src/main/java/interfacesData/IDataTable.java](#)

7.40 guiMain.IpCell Class Reference

[IpCell](#), Display a list of Ips on a list that can be removed with a button.

Inheritance diagram for guiMain.IpCell:



Public Member Functions

- [IpCell](#) ()
Class constructor.

Protected Member Functions

- void [updateItem](#) (String item, boolean empty)
Update the content of a Cell : set a style according to the parameters of a game.

Package Attributes

- HBox [hbox](#) = new HBox()
- Label [ipLabel](#) = new Label("")
- Pane [pane](#) = new Pane()
- Button [removeButton](#) = new Button("")

7.40.1 Detailed Description

[IpCell](#), Display a list of Ips on a list that can be removed with a button.

Author

IHM-MAIN Module

7.40.2 Constructor & Destructor Documentation

7.40.2.1 [guiMain.IpCell.IpCell](#) ()

Class constructor.

7.40.3 Member Function Documentation

7.40.3.1 void [guiMain.IpCell.updateItem](#) (String *item*, boolean *empty*) [protected]

Update the content of a Cell : set a style according to the parameters of a game.

Parameters

<i>item</i>	: the new content of a cell to be displayed
<i>empty</i>	: true if the cell is empty, false otherwise

7.40.4 Member Data Documentation

7.40.4.1 `HBox guiMain.IpCell.hbox = new HBox()` [package]

7.40.4.2 `Label guiMain.IpCell.ipLabel = new Label("")` [package]

7.40.4.3 `Pane guiMain.IpCell.pane = new Pane()` [package]

7.40.4.4 `Button guiMain.IpCell.removeButton = new Button("")` [package]

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

- [src/main/java/guiMain/IpCell.java](#)

7.41 guiMain.controller.IpConfigController Class Reference

This class implements the controller of the Ip Configuration page.

Public Member Functions

- void [setMainController](#) ([GuiMainController](#) c)
Set [GuiMainController](#) mainController.
- void [init](#) ()
Called at initialization.

Private Member Functions

- void [initIpsList](#) ()
Initialize the Ips list with the already available Ips.
- void [addIp](#) ()
Access an Ip to the list view.
- void [validate](#) ([ActionEvent](#) event)
Set the new Ip List and close the current window.

Private Attributes

- [GuiMainController](#) [mainController](#)
- [ListView](#)< [String](#) > [ipsListView](#)
- [TextField](#) [ipTextField](#)
- [Button](#) [addButton](#)
- [Button](#) [validateButton](#)
- [TextField](#) [port](#)

7.41.1 Detailed Description

This class implements the controller of the Ip Configuration page.

Author

IHM-Main module

7.41.2 Member Function Documentation

7.41.2.1 void guiMain.controller.IpConfigController.addIp () [private]

Access an Ip to the list view.

7.41.2.2 void guiMain.controller.IpConfigController.init ()

Called at initialization.

7.41.2.3 void guiMain.controller.IpConfigController.initIpsList () [private]

Initialize the Ips list with the already available Ips.

7.41.2.4 void guiMain.controller.IpConfigController.setMainController (GuiMainController *c*)

Set [GuiMainController](#) mainController.

Parameters

<i>c</i>	: GuiMainController
----------	-------------------------------------

7.41.2.5 void guiMain.controller.IpConfigController.validate (*ActionEvent event*) [private]

Set the new Ip List and close the current window.

Parameters

<i>event</i>	: validateButton
--------------	----------------------------------

7.41.3 Member Data Documentation

7.41.3.1 Button guiMain.controller.IpConfigController.addButton [private]

7.41.3.2 ListView<String> guiMain.controller.IpConfigController.ipsListView [private]

7.41.3.3 TextField guiMain.controller.IpConfigController.ipTextField [private]

7.41.3.4 GuiMainController guiMain.controller.IpConfigController.mainController [private]

7.41.3.5 TextField guiMain.controller.IpConfigController.port [private]

7.41.3.6 Button guiMain.controller.IpConfigController.validateButton [private]

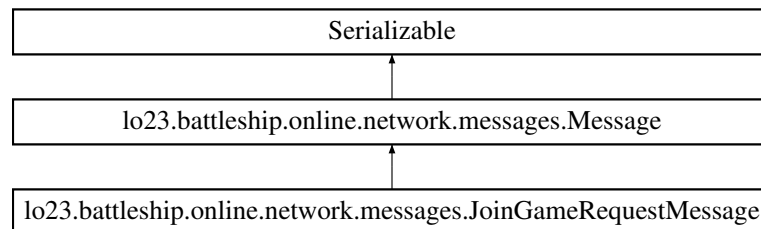
The documentation for this class was generated from the following file:

- [src/main/java/guiMain/controller/IpConfigController.java](#)

7.42 `lo23.battleship.online.network.messages.JoinGameRequestMessage` Class Reference

This class implements the message which is sent when when a user wants to join a game created by another user.

Inheritance diagram for `lo23.battleship.online.network.messages.JoinGameRequestMessage`:



Public Member Functions

- `JoinGameRequestMessage` (`Profile` senderGame, `Game` gameSend)
Allocates a new.
- `String` `getType` ()
Returns the type of the message.
- `void` `process` (`IDataCom` IData, `InetAddress` senderAddress)
Add the user `sender` to the game if possible using `IDataCom` interface.

Package Attributes

- `Profile` sender
- `Game` game

7.42.1 Detailed Description

This class implements the message which is sent when when a user wants to join a game created by another user.

This class extends the abstract `Message` and implements the two abstract methods: `getType` and `process`

See also

[Message](#)

Author

COM Module

7.42.2 Constructor & Destructor Documentation

7.42.2.1 `lo23.battleship.online.network.messages.JoinGameRequestMessage.JoinGameRequestMessage` (`Profile` senderGame, `Game` gameSend)

Allocates a new.

`JoinGameRequestMessage`

object.

Parameters

<i>senderGame</i>	: the User who wants to join the game.
<i>gameSend</i>	: the game that the user <i>senderGame</i> wants to join.

7.42.3 Member Function Documentation

7.42.3.1 String lo23.battleship.online.network.messages.JoinGameRequestMessage.getType ()

Returns the type of the message.

Implementation of an abstract method.

Returns

type :

String

7.42.3.2 void lo23.battleship.online.network.messages.JoinGameRequestMessage.process (IDataCom IData, InetAddress senderAddress)

Add the user *sender* to the game if possible using IDataCom interface.

Parameters

<i>IData</i>	: IDataCom instance of IDataCom interface.
<i>senderAddress</i>	: InetAddress sender's IP address

7.42.4 Member Data Documentation

7.42.4.1 Game lo23.battleship.online.network.messages.JoinGameRequestMessage.game [package]

7.42.4.2 Profile lo23.battleship.online.network.messages.JoinGameRequestMessage.sender [package]

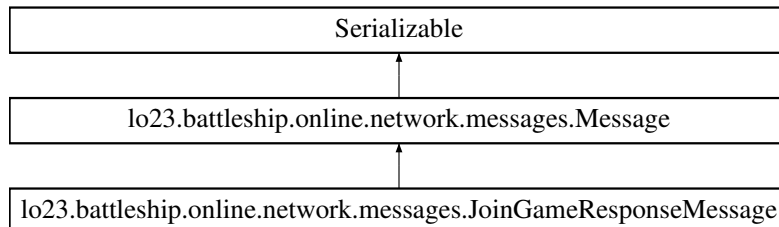
The documentation for this class was generated from the following file:

- [src/main/java/lo23/battleship/online/network/messages/JoinGameRequestMessage.java](#)

7.43 `lo23.battleship.online.network.messages.JoinGameResponseMessage` Class Reference

This class implements the message which is sent when when a creator of a game adds a second player to the game.

Inheritance diagram for `lo23.battleship.online.network.messages.JoinGameResponseMessage`:



Public Member Functions

- [JoinGameResponseMessage](#) (Boolean `isOk`, [Profile](#) `sender`, [Game](#) `game`)
Allocates a new.
- String [getType](#) ()
Returns the type of the message.
- void [process](#) ([IDataCom](#) `IData`, [InetAddress](#) `senderAddress`)
Notifies the user receiving this message that he has (or has not) joined the game.

Private Attributes

- [Game](#) `game`
- [Profile](#) `sender`
- Boolean `isOk`

Additional Inherited Members

7.43.1 Detailed Description

This class implements the message which is sent when when a creator of a game adds a second player to the game.

The creator of the game sends this message to the user who was added as second player.

This class extends the abstract [Message](#) and implements the two abstract methods: `getType` and `process`

See also

[Message](#)

Author

COM Module

7.43.2 Constructor & Destructor Documentation

7.43.2.1 Io23.battleship.online.network.messages.JoinGameResponseMessage.JoinGameResponseMessage (Boolean *isOk*, Profile *sender*, Game *game*)

Allocates a new.

[JoinGameResponseMessage](#)

object.

Parameters

<i>isOk</i>	is the answer depending on whether the opponent slot is available for the distant player willing to join (true : the slot is available, false : the slot isn't available).
<i>sender</i>	is the User class of the local player.
<i>game</i>	is the game which is willed to be joined by the distant player.

7.43.3 Member Function Documentation

7.43.3.1 String Io23.battleship.online.network.messages.JoinGameResponseMessage.getType ()

Returns the type of the message.

Implementation of an abstract method.

Returns

type :

`String`

7.43.3.2 void Io23.battleship.online.network.messages.JoinGameResponseMessage.process (IDataCom *IData*, InetAddress *senderAddress*)

Notifies the user receiving this message that he has (or has not) joined the game.

Parameters

<i>IData</i>	: <code>IDataCom</code> instance of IDataCom interface.
<i>senderAddress</i>	: <code>InetAddress</code> sender's IP address

7.43.4 Member Data Documentation

7.43.4.1 **Game** `lo23.battleship.online.network.messages.JoinGameResponseMessage.game` [private]

7.43.4.2 **Boolean** `lo23.battleship.online.network.messages.JoinGameResponseMessage.isOk` [private]

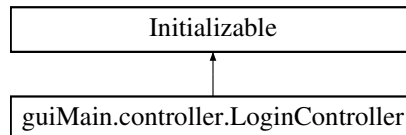
7.43.4.3 **Profile** `lo23.battleship.online.network.messages.JoinGameResponseMessage.sender` [private]

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

- `src/main/java/lo23/battleship/online/network/messages/JoinGameResponseMessage.java`

7.44 guiMain.controller.LoginController Class Reference

Inheritance diagram for `guiMain.controller.LoginController`:



Public Member Functions

- void `initialize` (URL location, ResourceBundle resources)
- void `setMainController` (`GuiMainController` c)
Set mainController to open the windows.

Package Functions

- void `login` (`ActionEvent` event)
Login of users in the system.
- void `openCreateAccountWindow` (`ActionEvent` event)
Open create account window.

Private Attributes

- `TextField` `userLogin`
- `PasswordField` `userPassword`
- `Label` `errorMessage`
- `Button` `loginButton`
- `Button` `createAccount`
- `GuiMainController` `mainController`

7.44.1 Member Function Documentation

7.44.1.1 void `guiMain.controller.LoginController.initialize` (URL location, ResourceBundle resources)

7.44.1.2 void `guiMain.controller.LoginController.login` (ActionEvent event) [package]

Login of users in the system.

Parameters

<i>event</i>	: loginButton event
--------------	-------------------------------------

7.44.1.3 void `guiMain.controller.LoginController.openCreateAccountWindow (ActionEvent event)` [`package`]

Open create account window.

Parameters

<i>event</i>	: createAccount button event
--------------	--

7.44.1.4 void `guiMain.controller.LoginController.setMainController (GuiMainController c)`

Set mainController to open the windows.

Parameters

<i>c</i>	: GuiMainController
----------	-------------------------------------

7.44.2 Member Data Documentation

7.44.2.1 Button `guiMain.controller.LoginController.createAccount` [`private`]

7.44.2.2 Label `guiMain.controller.LoginController.errorMessage` [`private`]

7.44.2.3 Button `guiMain.controller.LoginController.loginButton` [`private`]

7.44.2.4 [GuiMainController](#) `guiMain.controller.LoginController.mainController` [`private`]

7.44.2.5 TextField `guiMain.controller.LoginController.userLogin` [`private`]

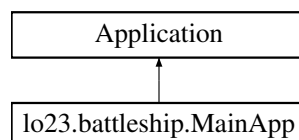
7.44.2.6 PasswordField `guiMain.controller.LoginController.userPassword` [`private`]

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

- `src/main/java/guiMain/controller/LoginController.java`

7.45 Io23.battleship.MainApp Class Reference

Inheritance diagram for `Io23.battleship.MainApp`:



Public Member Functions

- void [start](#) (Stage stage) throws Exception

Static Public Member Functions

- static void [main](#) (String[] args)
The [main\(\)](#) method is ignored in correctly deployed JavaFX application.

7.45.1 Member Function Documentation

7.45.1.1 static void `lo23.battleship.MainApp.main (String[] args)` `[static]`

The [main\(\)](#) method is ignored in correctly deployed JavaFX application.

[main\(\)](#) serves only as fallback in case the application can not be launched through deployment artifacts, e.g., in IDEs with limited FX support. NetBeans ignores [main\(\)](#).

Parameters

<code>args</code>	the command line arguments
-------------------	----------------------------

7.45.1.2 void `lo23.battleship.MainApp.start (Stage stage)` throws Exception

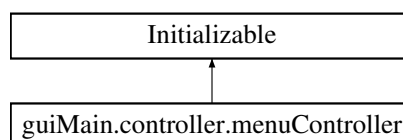
The documentation for this class was generated from the following file:

- `src/main/java/lo23/battleship/MainApp.java`

7.46 guiMain.controller.menuController Class Reference

This class call the lobby view and enable interaction between button and method.

Inheritance diagram for `guiMain.controller.menuController`:



Public Member Functions

- void [setMainController](#) ([GuiMainController](#) c)
Set [GuiMainController](#) mainController.
- void [init](#) ()
Init listView configuration.
- void [joinGame](#) ([Game](#) game)
Access game as player.
- void [lookGame](#) ([Game](#) game)
Access game as spectator.
- void [lookUser](#) ([User](#) user)
Display profile of others users.
- void [addUser](#) ([User](#) user)
Adds the user passed as a parameter to the list of users.
- void [removeUser](#) ([User](#) user)
Remove the user passed as a parameter to the list of users.
- void [addGame](#) ([Game](#) game)
Add the game passed as a parameter to the list of games.
- void [updateGameStatus](#) ([Game](#) game)
Update the game passed as a parameter in the list of games.
- void [removeGame](#) ([Game](#) removedGame)
Remove the game passed as a parameter to the list of games.
- void [initialize](#) (URL location, ResourceBundle resources)
Unused but necessary method.

Package Functions

- void [openChangeProfileWindow](#) (ActionEvent event)
Display changeProfile window.

Private Member Functions

- void [initUserList](#) ()
Initialise the user list in the main GUI.
- void [initGamesList](#) ()
Init the game list and attach it to an observable list.
- void [setImage](#) ()
Display the user's avatar on the view.
- void [option](#) ()
Access to option windows.
- void [disconnection](#) ()
Send disconnection message to other users and display the login view.
- void [refresh](#) ()
Explicit ask to receive informations again.
- void [openCreateGameWindow](#) (ActionEvent event) throws IOException
Open window to create new game.

Private Attributes

- [GuiMainController](#) `mainController`
- [ImageView](#) `avatarImage`
- [ListView< User >](#) `playersView`
- [ListView< Game >](#) `gamesView`
- [Button](#) `optionButton`
- [Button](#) `modifyProfileButton`
- [Button](#) `refreshButton`
- [Label](#) `warningLabel`

7.46.1 Detailed Description

This class call the lobby view and enable interaction between button and method.

Author

IHM-Main module

7.46.2 Member Function Documentation

7.46.2.1 void `guiMain.controller.menuController.addGame (Game game)`

Add the game passed as a parameter to the list of games.

Parameters

<i>game</i>	: game to add to the list.
-------------	----------------------------

7.46.2.2 void `guiMain.controller.menuController.addUser (User user)`

Adds the user passed as a parameter to the list of users.

Parameters

<i>user</i>	: user to add to the list.
-------------	----------------------------

7.46.2.3 void `guiMain.controller.menuController.disconnect ()` `[private]`

Send disconnection message to other users and display the login view.

7.46.2.4 void `guiMain.controller.menuController.init ()`

Init listView configuration.

7.46.2.5 void guiMain.controller.menuController.initGamesList () [private]

Init the game list and attach it to an observable list.

7.46.2.6 void guiMain.controller.menuController.initialize (URL *location*, ResourceBundle *resources*)

Unused but necessary method.

7.46.2.7 void guiMain.controller.menuController.initUserList () [private]

Initialise the user list in the main GUI.

7.46.2.8 void guiMain.controller.menuController.joinGame (Game *game*)

Access game as player.

Parameters

<i>game</i>	: game to access.
-------------	-------------------

7.46.2.9 void guiMain.controller.menuController.lookGame (Game *game*)

Access game as spectator.

Parameters

<i>game</i>	: game to access.
-------------	-------------------

7.46.2.10 void guiMain.controller.menuController.lookUser (User *user*)

Display profile of others users.

Parameters

<i>user</i>	: user's profil to display.
-------------	-----------------------------

7.46.2.11 void guiMain.controller.menuController.openChangeProfileWindow (ActionEvent *event*) [package]

Display changeProfile window.

Parameters

<i>event</i>	: modifyProfileButton event
--------------	---

7.46.2.12 void `guiMain.controller.menuController.openCreateGameWindow (ActionEvent event)` throws `IOException`
[private]

Open window to create new game.

Parameters

<i>event</i>	: button #createGame event click
--------------	----------------------------------

Exceptions

<i>IOException</i>	
--------------------	--

7.46.2.13 void `guiMain.controller.menuController.option ()` [private]

Access to option windows.

7.46.2.14 void `guiMain.controller.menuController.refresh ()` [private]

Explicit ask to receive informations again.

7.46.2.15 void `guiMain.controller.menuController.removeGame (Game removedGame)`

Remove the game passed as a parameter to the list of games.

Parameters

<i>game</i>	: game to remove to the list.
-------------	-------------------------------

7.46.2.16 void `guiMain.controller.menuController.removeUser (User user)`

Remove the user passed as a parameter to the list of users.

Parameters

<i>user</i>	: user to remove to the list.
-------------	-------------------------------

7.46.2.17 void guiMain.controller.menuController.setImage () [private]

Display the user's avatar on the view.

7.46.2.18 void guiMain.controller.menuController.setMainController (GuiMainController c)

Set [GuiMainController](#) mainController.

Parameters

<i>c</i>	: GuiMainController
----------	-------------------------------------

7.46.2.19 void guiMain.controller.menuController.updateGameStatus (Game game)

Update the game passed as a parameter in the list of games.

Parameters

<i>game</i>	: user to game in the list.
-------------	-----------------------------

7.46.3 Member Data Documentation

7.46.3.1 ImageView guiMain.controller.menuController.avatarImage [private]

7.46.3.2 ListView<Game> guiMain.controller.menuController.gamesView [private]

7.46.3.3 GuiMainController guiMain.controller.menuController.mainController [private]

7.46.3.4 Button guiMain.controller.menuController.modifyProfileButton [private]

7.46.3.5 Button guiMain.controller.menuController.optionButton [private]

7.46.3.6 ListView<User> guiMain.controller.menuController.playersView [private]

7.46.3.7 Button guiMain.controller.menuController.refreshButton [private]

7.46.3.8 Label guiMain.controller.menuController.warningLabel [private]

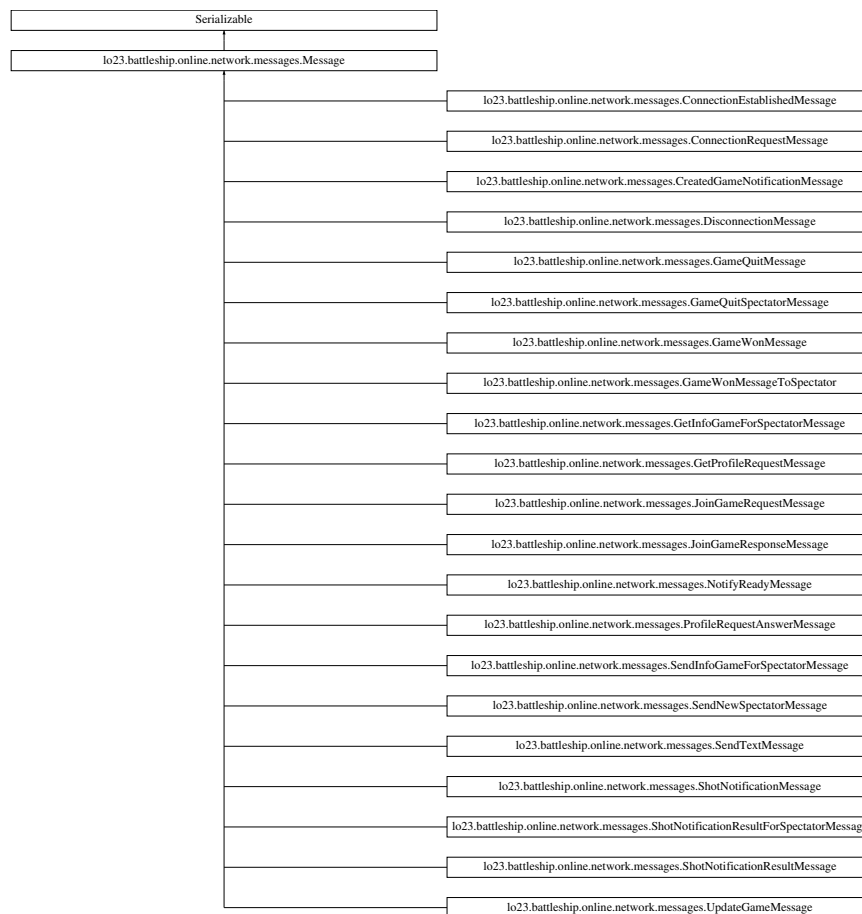
The documentation for this class was generated from the following file:

- src/main/java/guiMain/controller/[menuController.java](#)

7.47 Io23.battleship.online.network.messages.Message Class Reference

This abstract class implements the core structure of a message sent on the network.

Inheritance diagram for Io23.battleship.online.network.messages.Message:



Public Member Functions

- abstract String [getType](#) ()
Returns the type of the message.
- abstract void [process](#) (IDataCom IData, InetAddress senderAddress)
Abstract method, interacts with data package interface (IDataCom) once implemented.

Package Attributes

- String [type](#)

7.47.1 Detailed Description

This abstract class implements the core structure of a message sent on the network.

Every single message sent on the network has to extend this class. This class declares the two abstract methods: `getType` and `process`

Author

COM Module

7.47.2 Member Function Documentation

7.47.2.1 `abstract String lo23.battleship.online.network.messages.Message.getType ()` [abstract]

Returns the type of the message.

Abstract method.

Returns

type :

String

7.47.2.2 `abstract void lo23.battleship.online.network.messages.Message.process (IDataCom IData, InetAddress senderAddress)` [abstract]

Abstract method, interacts with data package interface (IDataCom) once implemented.

Parameters

<i>IData</i>	: IDataCom instance of IDataCom interface.
<i>senderAddress</i>	: InetAddress sender's IP address

7.47.3 Member Data Documentation

7.47.3.1 `String lo23.battleship.online.network.messages.Message.type` [package]

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

- `src/main/java/lo23/battleship/online/network/messages/Message.java`

7.48 structData.MessageType Enum Reference

Enumerate for the types of messages (Class Message)

Public Attributes

- `ERROR`
- `NOTIFICATION`

7.48.1 Detailed Description

Enumerate for the types of messages (Class Message)

7.48.2 Member Data Documentation

7.48.2.1 structData.MessageType.ERROR

7.48.2.2 structData.MessageType.NOTIFICATION

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

- [src/main/java/structData/MessageType.java](#)

7.49 Io23.battleship.online.network.NetworkController Class Reference

This class implements the network controller.

Public Member Functions

- `int getPort ()`
- `void sendMessage (Message message, InetAddress destinationIpAddress)`
Sends a message (.
- `List< InetAddress > getIPTable ()`
Returns the list of IP addresses of connected users.
- `void setDataInterface (IDataCom IData)`
Sets IDataCom instance for the network controller and the instance of COMInterface.
- `COMInterface getCOMInterface ()`
Returns the instance of the class which implements COMInterface.
- `List< InetAddress > filterUnknownIPAddresses (List< InetAddress > IPAddressesTable)`
Filter out unknown IP addresses.
- `List< InetAddress > filterKnownIPAddressesToNotify (List< InetAddress > IPAddressesTable)`
Filter out known IP addresses which are not in the submitted IP Address list Returns IP Addresses which are in the networkState and different than the local address but are not the parameter IPAddressesTable
- `void addToNetwork (User sender, InetAddress senderAddress, Game game)`
Add a new user with his corresponding IP address to the network state and to the list of users accessible in the entire application (data package).
- `void removeFromNetwork (User user, Game game)`
Removes a user from the network state and the list of connected users.
- `Set< User > getConnectedUsers ()`
Returns the collection of connected users.

Static Public Member Functions

- `static NetworkController getInstance ()`
Implementation of Singleton Design Pattern getInstance static method Returns the unique instance of NetworkController.

Package Functions

- void [launchServer](#) ()
Initializes and open a [NetworkServer](#) instance.
- InetAddress [getAddressForUser](#) (User user)
Finds the corresponding IP Address to a user.
- void [closeListener](#) ()
Close server so that the application stops receiving messages.
- InetAddress [removeUnreachableHost](#) (InetAddress address)
*Removes the user corresponding the IP address *address* when the address is unreachable.*
- void [clearNetwork](#) ()
Clear network state : remove all known players and their corresponding IP addresses.

Private Member Functions

- [NetworkController](#) ()
Allocates a new.
- boolean [addUserToNetwork](#) (User user, InetAddress senderAddress)
*Add a new user with his corresponding IP address to the network state, if the user is not already in the network state
Returns true if the user was added successfully and false otherwise return.*

Private Attributes

- int [port](#)
- [NetworkModuleInterface](#) [networkInterface](#)
- HashMap< [User](#), InetAddress > [networkState](#)
- [IDataCom](#) [dataInterface](#)
- [NetworkServer](#) [networkServer](#)

Static Private Attributes

- static [NetworkController](#) [instance](#)

7.49.1 Detailed Description

This class implements the network controller.

It contains references to all main components in the network package, and the network state (HashMap of all currently connected users)

It is based on Singleton Design Pattern

Author

COM Module

7.49.2 Constructor & Destructor Documentation

7.49.2.1 `lo23.battleship.online.network.NetworkController.NetworkController ()` [private]

Allocates a new.

`NetworkController`

object. Initializes `networkState` and `networkInterface`

7.49.3 Member Function Documentation

7.49.3.1 `void lo23.battleship.online.network.NetworkController.addToNetwork (User sender, InetAddress senderAddress, Game game)`

Add a new user with his corresponding IP address to the network state and to the list of users accessible in the entire application (data package).

Optionally it adds the game the new user created.

Parameters

<code>sender</code>	
---------------------	--

7.49.3.2 `boolean lo23.battleship.online.network.NetworkController.addUserToNetwork (User user, InetAddress senderAddress)` [private]

Add a new user with his corresponding IP address to the network state, if the user is not already in the network state Returns true if the user was added successfully and false otherwise return.

`boolean`

7.49.3.3 `void lo23.battleship.online.network.NetworkController.clearNetwork ()` [package]

Clear network state : remove all known players and their corresponding IP addresses.

7.49.3.4 `void lo23.battleship.online.network.NetworkController.closeListener ()` [package]

Close server so that the application stops receiving messages.

7.49.3.5 `List<InetAddress> lo23.battleship.online.network.NetworkController.filterKnownIPAddressesToNotify (List<InetAddress> > iPAddressesTable)`

Filter out known IP addresses which are not in the submitted IP Address list Returns IP Addresses which are in the networkState and different than the local address but are not the parameter `iPAddressesTable`

Returns

`List<InetAddress>`

7.49.3.6 `List<InetAddress> lo23.battleship.online.network.NetworkController.filterUnknownIPAddresses (List<InetAddress> > iPAddressesTable)`

Filter out unknown IP addresses.

Returns IP Addresses of the parameter `iPAddressesTable` which are not in the networkState and different than the local address

Returns

`List<InetAddress>`

7.49.3.7 `InetAddress lo23.battleship.online.network.NetworkController.getAddressForUser (User user)` [package]

Finds the corresponding IP Address to a user.

Parameters

<i>user</i>	:
	User
	user of which the IP"address is needed

Returns

`InetAddress`

7.49.3.8 `COMInterface lo23.battleship.online.network.NetworkController.getCOMInterface ()`

Returns the instance of the class which implements [COMInterface](#).

Returns

`COMInterface`

7.49.3.9 Set<User> io23.battleship.online.network.NetworkController.getConnectedUsers ()

Returns the collection of connected users.

Returns

Set<User>

7.49.3.10 static NetworkController io23.battleship.online.network.NetworkController.getInstance () [static]

Implementation of Singleton Design Pattern `getInstance` static method Returns the unique instance of [NetworkController](#).

Returns

[NetworkController](#)

7.49.3.11 List<InetAddress> io23.battleship.online.network.NetworkController.getIPTable ()

Returns the list of IP addresses of connected users.

Returns

List<InetAddress>

7.49.3.12 int io23.battleship.online.network.NetworkController.getPort ()

7.49.3.13 void io23.battleship.online.network.NetworkController.launchServer () [package]

Initializes and open a [NetworkServer](#) instance.

Sets the `IDataCom` instance accordingly

7.49.3.14 void io23.battleship.online.network.NetworkController.removeFromNetwork (User user, Game game)

Removes a user from the network state and the list of connected users.

Optionally it removes the game the user might have created

7.49.3.15 InetAddress io23.battleship.online.network.NetworkController.removeUnreachableHost (InetAddress address) [package]

Removes the user corresponding the IP address `address` when the address is unreachable.

Returns the IP address of the removed user

Returns

InetAddress

7.49.3.16 `void io23.battleship.online.network.NetworkController.sendMessage (Message message, InetAddress destinationIpAddress)`

Sends a message (.

`Message`

) to a destination host (`InetAddress`) using a

`NetworkSender`

thread

Parameters

<i>message</i>	: Message message to be sent
<i>destinationIpAddress</i>	: InetAddress destination IP Address

7.49.3.17 void `lo23.battleship.online.network.NetworkController.setDataInterface (IDataCom IData)`

Sets IDataCom instance for the network controller and the instance of [COMInterface](#).

Parameters

<i>IData</i>	: IDataCom instance of IDataCom
--------------	---------------------------------------

7.49.4 Member Data Documentation

7.49.4.1 `IDataCom lo23.battleship.online.network.NetworkController.dataInterface` [private]

7.49.4.2 `NetworkController lo23.battleship.online.network.NetworkController.instance` [static], [private]

7.49.4.3 `NetworkModuleInterface lo23.battleship.online.network.NetworkController.networkInterface` [private]

7.49.4.4 `NetworkServer lo23.battleship.online.network.NetworkController.networkServer` [private]

7.49.4.5 `HashMap<User, InetAddress> lo23.battleship.online.network.NetworkController.networkState` [private]

7.49.4.6 `int lo23.battleship.online.network.NetworkController.port` [private]

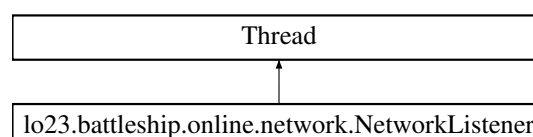
The documentation for this class was generated from the following file:

- `src/main/java/lo23/battleship/online/network/NetworkController.java`

7.50 `lo23.battleship.online.network.NetworkListener` Class Reference

This class implements network listeners.

Inheritance diagram for `lo23.battleship.online.network.NetworkListener`:



Public Member Functions

- void [run](#) ()

Package Functions

- [NetworkListener](#) ([NetworkServer](#) server, [ServerSocket](#) socket)
Allocates a new.
- void [setIsRunning](#) (boolean newValue)
Sets isRunning with a new value.
- boolean [getIsRunning](#) ()
Returns a boolean indicating if the listener is running.
- [InetAddress](#) [getServerSocketIPAddress](#) ()
Returns server socket IP address as InetAddress Object.
- void [closeSocket](#) ()
- void [setDataInterface](#) ([IDataCom](#) IData)
Sets the IDataCom interface instance.

Private Member Functions

- [Message](#) [read](#) () throws [IOException](#)
Read and deserialize object of type Message transferred through the network.

Private Attributes

- [ServerSocket](#) [serverSocket](#) = null
- boolean [isRunning](#)
- [IDataCom](#) [dataInterface](#)
- [ObjectInputStream](#) [reader](#)
- [NetworkServer](#) [server](#)

7.50.1 Detailed Description

This class implements network listeners.

Network listeners use and encapsulate server sockets to receive messages transferred on the network between players, object input streams to read the messages and then run the process associated with each of those message. The process of each message sometimes requires the [IDataCom](#) interface instance.

Author

COM Module

See also

[java.lang.Thread](#)

[Message](#)

[java.io.ObjectInputStream](#) Plus it extends [Thread](#) class so that it is [run](#) asynchronously from the thread of the main application.

7.50.2 Constructor & Destructor Documentation

7.50.2.1 `lo23.battleship.online.network.NetworkListener.NetworkListener (NetworkServer server, ServerSocket socket)` [package]

Allocates a new.

`NetworkListener`

object.

Parameters

<code>server</code>	: <code>NetworkServer</code> <code>NetworkServer</code> instance encapsulating <code>NetworkListener</code>
<code>socket</code>	: <code>ServerSocket</code> socket receiving network message

7.50.3 Member Function Documentation

7.50.3.1 `void lo23.battleship.online.network.NetworkListener.closeSocket ()` [package]

7.50.3.2 `boolean lo23.battleship.online.network.NetworkListener.getIsRunning ()` [package]

Returns a boolean indicating if the listener is running.

Returns

`isRunning` :

`boolean`

7.50.3.3 `InetAddress lo23.battleship.online.network.NetworkListener.getServerSocketIPAddress ()` [package]

Returns server socket IP address as `InetAddress` Object.

Returns

the server socket host :

`InetAddress`

7.50.3.4 Message lo23.battleship.online.network.NetworkListener.read () throws IOException [private]

Read and deserialize object of type Message transferred through the network.

Returns

the read message :

Message

7.50.3.5 void lo23.battleship.online.network.NetworkListener.run ()**7.50.3.6 void** lo23.battleship.online.network.NetworkListener.setDataInterface (IDataCom IData) [package]

Sets the IDataCom interface instance.

Parameters

<i>IData</i>	:
	IDataCom
	instance of IDataCom interface

7.50.3.7 void lo23.battleship.online.network.NetworkListener.setIsRunning (boolean newValue) [package]

Sets isRunning with a new value.

Parameters

<i>newValue</i>	:
	boolean
	new value of isRunning

7.50.4 Member Data Documentation**7.50.4.1 IDataCom** lo23.battleship.online.network.NetworkListener.dataInterface [private]**7.50.4.2 boolean** lo23.battleship.online.network.NetworkListener.isRunning [private]**7.50.4.3 ObjectInputStream** lo23.battleship.online.network.NetworkListener.reader [private]**7.50.4.4 NetworkServer** lo23.battleship.online.network.NetworkListener.server [private]**7.50.4.5 ServerSocket** lo23.battleship.online.network.NetworkListener.serverSocket = null [private]

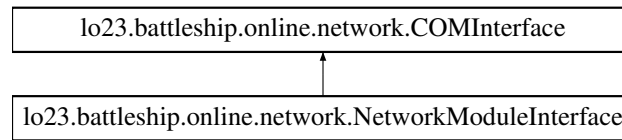
The documentation for this class was generated from the following file:

- src/main/java/lo23/battleship/online/network/[NetworkListener.java](#)

7.51 io23.battleship.online.network.NetworkModuleInterface Class Reference

This class implements the [COMInterface](#) interface and the different methods (services) it offers.

Inheritance diagram for io23.battleship.online.network.NetworkModuleInterface:



Public Member Functions

- void [notifyReady](#) ([User](#) user, [Player](#) playerToNotify)
Notifies if a player is ready or not.
- void [sendMessage](#) ([ChatMessage](#) chatMessage, [Game](#) g)
Sends a chat message.
- void [getProfile](#) ([User](#) userRequested)
Retrieves an user profile.
- void [notifyJoinGameResponse](#) (boolean isOk, [Profile](#) user, [Game](#) game)
allow an user to join a game
- void [changeStatusGame](#) ([Game](#) game)
Update game object (containing game.status)
- void [notifyNewGame](#) ([Game](#) g)
notify a new game
- void [joinGame](#) ([Game](#) g)
allow an user to join a game
- void [askDisconnection](#) ()
allow an user to be disconnected to the network
- void [sendShot](#) ([Player](#) player, [Game](#) game, [Shot](#) shot)
send a shot from a player on the right game
- void [searchForPlayers](#) ()
Initiates network discovery.
- void [notifyGameWon](#) ()
notifies every user to remove the game from their list
- void [coordinates](#) ([Player](#) destPlayer, [Shot](#) resultShot, [Game](#) game, [Boat](#) boat)
send a shot result to a player on the right game
- void [removeGame](#) ([Game](#) game)
notifies every user to remove the game from their list
- void [getInfoGameForSpectator](#) ([Player](#) player, [User](#) spec)
spectator send a request to a player in order to have the game informations
- void [sendInfoGameForSpectator](#) ([Game](#) game, [User](#) spec)
player send the game to the spectator after his request
- void [sendNewSpectator](#) ([User](#) u, [Player](#) p, [HashSet](#)< [User](#) > listSpectators)
alert everybody (the other player and the spectator) that there is a new spectator
- void [gameQuitSpectator](#) ([User](#) spec, [Game](#) game)
alert everybody that the spectator quit the game
- void [clearNetwork](#) ()
Clears network state : removes all players from local user list of connected players.

Package Functions

- [NetworkModuleInterface](#) ([NetworkController](#) cont)
Allocates new.
- void [setDataInterface](#) ([IDataCom](#) IData)
Sets the IDataCom instance (to call IDataCom services(methods))

Private Attributes

- [IDataCom](#) dataInterface
- [NetworkController](#) controller

7.51.1 Detailed Description

This class implements the [COMInterface](#) interface and the different methods (services) it offers.

Author

COM Module

See also

[COMInterface](#)

7.51.2 Constructor & Destructor Documentation

7.51.2.1 Io23.battleship.online.network.NetworkModuleInterface.NetworkModuleInterface ([NetworkController](#) cont) [package]

Allocates new.

[NetworkModuleInterface](#)

object

Parameters

<i>cont</i>	:
	NetworkController
	Network Controller instance

7.51.3 Member Function Documentation

7.51.3.1 void `lo23.battleship.online.network.NetworkModuleInterface.askDisconnection ()`

allow an user to be disconnected to the network

Implements [lo23.battleship.online.network.COMInterface](#).

7.51.3.2 void `lo23.battleship.online.network.NetworkModuleInterface.changeStatusGame (Game game)`

Update game object (containing game.status)

Parameters

<i>game</i>	: Game of which the status changed
-------------	------------------------------------

Implements [lo23.battleship.online.network.COMInterface](#).

7.51.3.3 void `lo23.battleship.online.network.NetworkModuleInterface.clearNetwork ()`

Clears network state : removes all players from local user list of connected players.

Implements [lo23.battleship.online.network.COMInterface](#).

7.51.3.4 void `lo23.battleship.online.network.NetworkModuleInterface.coordinates (Player destPlayer, Shot resultShot, Game game, Boat boat)`

send a shot result to a player on the right game

Parameters

<i>destPlayer</i>	destination player
<i>game</i>	where the ships are
<i>resultShot</i>	result of the shot
<i>boat</i>	optional

Implements [lo23.battleship.online.network.COMInterface](#).

7.51.3.5 void `lo23.battleship.online.network.NetworkModuleInterface.gameQuitSpectator (User spec, Game game)`

alert everybody that the spectator quit the game

Parameters

<i>spec</i>	spectator who quit the game
<i>game</i>	game quit by the spectator (permit to retrieve player and listSpectator)

Implements [lo23.battleship.online.network.COMInterface](#).

7.51.3.6 void lo23.battleship.online.network.NetworkModuleInterface.getInfoGameForSpectator (Player *player*, User *spec*)

spectator send a request to a player in order to have the game informations

Parameters

<i>player</i>	player game
<i>spec</i>	spectator who ask for request

Implements [lo23.battleship.online.network.COMInterface](#).

7.51.3.7 void lo23.battleship.online.network.NetworkModuleInterface.getProfile (User *user*)

Retrieves an user profile.

Parameters

<i>user</i>	: User of which local user asks for profile
-------------	---

Implements [lo23.battleship.online.network.COMInterface](#).

7.51.3.8 void lo23.battleship.online.network.NetworkModuleInterface.joinGame (Game *g*)

allow an user to join a game

Parameters

<i>g</i>	Game the local user wants to join
----------	-----------------------------------

Implements [lo23.battleship.online.network.COMInterface](#).

7.51.3.9 void lo23.battleship.online.network.NetworkModuleInterface.notifyGameWon ()

notifies every user to remove the game from their list

Implements [lo23.battleship.online.network.COMInterface](#).

7.51.3.10 void lo23.battleship.online.network.NetworkModuleInterface.notifyJoinGameResponse (boolean *isOk*, Profile *user*, Game *g*)

allow an user to join a game

Parameters

<i>isOk</i>	access to <code>game</code> true=access granted false= access denied
<i>user</i>	who asked to join the game
<i>g</i>	Game <code>user</code> joined if <code>isOk</code>

Implements [lo23.battleship.online.network.COMInterface](#).

7.51.3.11 void [lo23.battleship.online.network.NetworkModuleInterface](#).notifyNewGame (Game *g*)

notify a new game

Parameters

<i>g</i>	: newly created game with one player
----------	--------------------------------------

Implements [lo23.battleship.online.network.COMInterface](#).

7.51.3.12 void [lo23.battleship.online.network.NetworkModuleInterface](#).notifyReady (User *user*, Player *playerToNotify*)

Notifies if a player is ready or not.

Parameters

<i>user</i>	player who is notify
-------------	----------------------

Implements [lo23.battleship.online.network.COMInterface](#).

7.51.3.13 void [lo23.battleship.online.network.NetworkModuleInterface](#).removeGame (Game *game*)

notifies every user to remove the game from their list

Parameters

<i>game</i>	Game to delete
-------------	----------------

Implements [lo23.battleship.online.network.COMInterface](#).

7.51.3.14 void [lo23.battleship.online.network.NetworkModuleInterface](#).searchForPlayers ()

Initiates network discovery.

Search for players who are connected

Implements [lo23.battleship.online.network.COMInterface](#).

7.51.3.15 void lo23.battleship.online.network.NetworkModuleInterface.sendMessage (*ChatMessage chatMessage*, *Game g*)

Sends a chat message.

Parameters

<i>chatMessage</i>	message send
<i>g</i>	: game related to the message

Implements [lo23.battleship.online.network.COMInterface](#).

7.51.3.16 void lo23.battleship.online.network.NetworkModuleInterface.sendInfoGameForSpectator (*Game game*, *User spec*)

player send the game to the spectator after his request

Parameters

<i>game</i>	game send to the spectator
<i>spec</i>	spectator who want to see the game

Implements [lo23.battleship.online.network.COMInterface](#).

7.51.3.17 void lo23.battleship.online.network.NetworkModuleInterface.sendNewSpectator (*User u*, *Player p*, *HashSet<User> listSpectator*)

alert everybody (the other player and the spectator) that there is a new spectator

Parameters

<i>u</i>	new spectator
<i>p</i>	other player
<i>listSpectator</i>	list of all actual spectators

Implements [lo23.battleship.online.network.COMInterface](#).

7.51.3.18 void lo23.battleship.online.network.NetworkModuleInterface.sendShot (*Player player*, *Game g*, *Shot shot*)

send a shot from a player on the right game

Parameters

<i>player</i>	: <i>Player</i> who send the shot
---------------	---

Parameters

<i>g</i>	: Game where the ships are
<i>shot</i>	: Shot where the player shot

Implements [lo23.battleship.online.network.COMInterface](#).

7.51.3.19 void [lo23.battleship.online.network.NetworkModuleInterface.setDataInterface \(IDataCom IData \)](#) [package]

Sets the IDataCom instance (to call IDataCom services(methods))

7.51.4 Member Data Documentation

7.51.4.1 **NetworkController** [lo23.battleship.online.network.NetworkModuleInterface.controller](#) [private]

7.51.4.2 **IDataCom** [lo23.battleship.online.network.NetworkModuleInterface.dataInterface](#) [private]

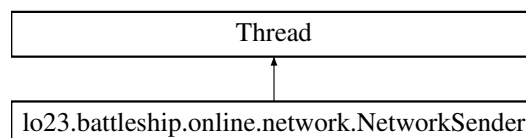
The documentation for this class was generated from the following file:

- [src/main/java/lo23/battleship/online/network/NetworkModuleInterface.java](#)

7.52 lo23.battleship.online.network.NetworkSender Class Reference

This class implements network senders.

Inheritance diagram for [lo23.battleship.online.network.NetworkSender](#):



Public Member Functions

- void [run](#) ()

Package Functions

- [NetworkSender](#) (InetAddress [host](#), int [port](#), [Message](#) [message](#))
Allocates a new.

Private Attributes

- InetAddress [host](#)
- int [port](#)
- Socket [sock](#)
- ObjectOutputStream [writer](#) = null
- [Message](#) [message](#)
- [NetworkController](#) [controller](#) = [NetworkController.getInstance\(\)](#)

7.52.1 Detailed Description

This class implements network senders.

Network senders are threads (Thread) responsible of sending message though the network. They use socket and output streams to connect to the destination host and send the serialized message.

See also

[java.lang.Thread](#)

7.52.2 Constructor & Destructor Documentation

7.52.2.1 [Io23.battleship.online.network.NetworkSender.NetworkSender](#) (InetAddress *host*, int *port*, [Message](#) *message*)
[package]

Allocates a new.

[NetworkSender](#)

object.

Parameters

<i>host</i>	: InetAddress destination host of the message
<i>port</i>	: int host's port on which the message sent
<i>message</i>	: Message message to be sent

7.52.3 Member Function Documentation

7.52.3.1 void lo23.battleship.online.network.NetworkSender.run ()

Overriding java.lang.Thread run method Connect to **host** on port **port** and send **message** through an output stream.

7.52.4 Member Data Documentation

7.52.4.1 NetworkController lo23.battleship.online.network.NetworkSender.controller = NetworkController.getInstance() [private]

7.52.4.2 InetAddress lo23.battleship.online.network.NetworkSender.host [private]

7.52.4.3 Message lo23.battleship.online.network.NetworkSender.message [private]

7.52.4.4 int lo23.battleship.online.network.NetworkSender.port [private]

7.52.4.5 Socket lo23.battleship.online.network.NetworkSender.sock [private]

7.52.4.6 ObjectOutputStream lo23.battleship.online.network.NetworkSender.writer = null [private]

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

- src/main/java/lo23/battleship/online/network/[NetworkSender.java](#)

7.53 lo23.battleship.online.network.NetworkServer Class Reference

This class implements the network server.

Public Member Functions

- void [open](#) () throws IOException
*Opens listener on **address** and the **port***
- void [close](#) ()
Close server (and listener accordingly)

Package Functions

- [NetworkServer](#) ([NetworkController](#) networkController, int port)
Allocates a new.
- void [setDataInterface](#) (IDataCom IData)
Sets IDataCom interface with the right IDataCom interface instance.
- InetAddress [getIpAddress](#) ()
Returns server (and thus listener) host IP address.
- boolean [isOpened](#) ()
Returns a boolean indicating if the server (and thus the listener) is opened (listening)

Private Attributes

- int [port](#)
- InetAddress [address](#)
- [NetworkListener](#) [listener](#) = null
- [NetworkController](#) [networkController](#)
- [IDataCom](#) [dataInterface](#)

Static Private Attributes

- static int [backlog](#) = 100

7.53.1 Detailed Description

This class implements the network server.

The network server is an encapsulation of the network listener. It is responsible of opening the network listener with the right IP address and closing the network listener when needed.

See also

[NetworkListener](#)
[IDataCom](#)

7.53.2 Constructor & Destructor Documentation

7.53.2.1 `Io23.battleship.online.network.NetworkServer.NetworkServer (NetworkController networkController, int port)`
[package]

Allocates a new.

[NetworkServer](#)

object. Finds the right IP address to use as host of the listener

Parameters

<i>networkController</i>	: <code>NetworkController</code> instance of NetworkController class
<i>port</i>	: <code>int</code> port on which the server (and the listener) is (are) opened

7.53.3 Member Function Documentation

7.53.3.1 void lo23.battleship.online.network.NetworkServer.close ()

Close server (and listener accordingly)

7.53.3.2 InetAddress lo23.battleship.online.network.NetworkServer.getIpAddress () [package]

Returns server (and thus listener) host IP address.

Returns

InetAddress

7.53.3.3 boolean lo23.battleship.online.network.NetworkServer.isOpened () [package]

Returns a boolean indicating if the server (and thus the listener) is opened (listening)

Returns

boolean

7.53.3.4 void lo23.battleship.online.network.NetworkServer.open () throws IOException

Opens listener on **address** and the **port**

7.53.3.5 void lo23.battleship.online.network.NetworkServer.setDataInterface (IDataCom IData) [package]

Sets IDataCom interface with the right IDataCom interface instance.

Parameters

<i>IData</i>	:
	IDataCom
	instance of IDataCom interface

7.53.4 Member Data Documentation

7.53.4.1 InetAddress lo23.battleship.online.network.NetworkServer.address [private]

7.53.4.2 int lo23.battleship.online.network.NetworkServer.backlog = 100 [static], [private]

7.53.4.3 `IDataCom lo23.battleship.online.network.NetworkServer.dataInterface` `[private]`

7.53.4.4 `NetworkListener lo23.battleship.online.network.NetworkServer.listener = null` `[private]`

7.53.4.5 `NetworkController lo23.battleship.online.network.NetworkServer.networkController` `[private]`

7.53.4.6 `int lo23.battleship.online.network.NetworkServer.port` `[private]`

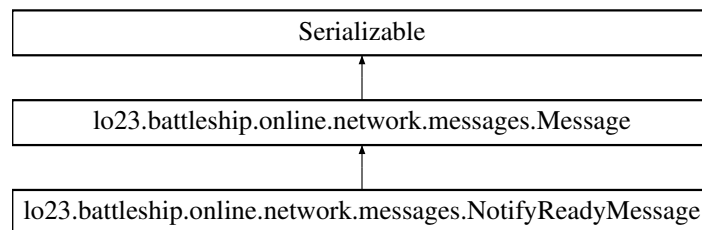
The documentation for this class was generated from the following file:

- `src/main/java/lo23/battleship/online/network/NetworkServer.java`

7.54 `lo23.battleship.online.network.messages.NotifyReadyMessage` Class Reference

This class implements the message sent by the local user (player) to the other player when he is ready to play.

Inheritance diagram for `lo23.battleship.online.network.messages.NotifyReadyMessage`:



Public Member Functions

- `NotifyReadyMessage (User sender, User destUser)`
Allocates a new.
- `String getType ()`
Returns the type of the message.
- `void process (IDataCom IData, InetAddress senderAddress)`
Notifies data package interface that the other player (sender of this message) is ready.

Private Attributes

- `User sender`
- `User destUser`

Additional Inherited Members

7.54.1 Detailed Description

This class implements the message sent by the local user (player) to the other player when he is ready to play.

See also

[Message](#)

Author

COM Module

7.54.2 Constructor & Destructor Documentation

7.54.2.1 `Io23.battleship.online.network.messages.NotifyReadyMessage.NotifyReadyMessage (User sender, User destUser)`

Allocates a new.

[NotifyReadyMessage](#)

object

Parameters

<i>sender</i>	
---------------	--

7.54.3 Member Function Documentation

7.54.3.1 `String Io23.battleship.online.network.messages.NotifyReadyMessage.getType ()`

Returns the type of the message.

Implementation of an abstract method.

Returns

type :

`String`

7.54.3.2 `void Io23.battleship.online.network.messages.NotifyReadyMessage.process (IDataCom IData, InetAddress senderAddress)`

Notifies data package interface that the other player (sender of this message) is ready.

Parameters

<i>IData</i>	: IDataCom instance of IDataCom interface.
<i>senderAddress</i>	: InetAddress sender's IP address

7.54.4 Member Data Documentation

7.54.4.1 User `lo23.battleship.online.network.messages.NotifyReadyMessage.destUser` [private]

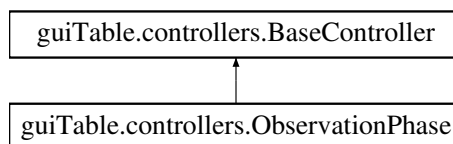
7.54.4.2 User `lo23.battleship.online.network.messages.NotifyReadyMessage.sender` [private]

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

- `src/main/java/lo23/battleship/online/network/messages/NotifyReadyMessage.java`

7.55 guiTable.controllers.ObservationPhase Class Reference

Inheritance diagram for guiTable.controllers.ObservationPhase:



Public Member Functions

- void `setTableController` (`GuiTableController` c)
Sets the `GuiTableController`.
- void `showVictory` (int player)
Shows victory message.
- void `logMsg` (String msg)
log message into interface.
- void `logYesNoMsg` (String msg)
log yesNoMessage into interface.
- void `setTurn` (Boolean t)
Sets the turn of the players and colors the boards.
- void `sunkPlayerBoat` (int i, `Boat` boat)
Calls another function to draw the sunked boat according to the parameters.
- `AnchorPane` `getChatPane` ()
- String `getLbj1` ()
- void `setLbj1` (String lbj1)
- String `getLbj2` ()
- void `setLbj2` (String lbj2)

Public Attributes

- boolean `turn` = false
- `HashMap< Integer, GridPane >` `tablePlayer`

Protected Member Functions

- void `placeShotTo` (`Shot` shot, `GridPane` grid)
Adds a shot on a board.
- void `sunkABoat` (`GridPane` grid, `Boat` boat)
Draws a sunked boat.

Protected Attributes

- `GuiTableController` `tableController`
- final String `STYLE_MY_TURN` = "-fx-background-color: #FFFFFF;";
- final String `STYLE_OTHER_TURN` = "-fx-background-color: #EEEEEE;";
- String `MY_TURN_MSG` = "A votre tour de jouer, cliquer sur une case puis sur le bouton valider"
- String `OTHER_TURN_MSG` = "Au tour de l'adversaire de jouer, merci de patienter"
- String `EXIT_GAME_MSG` = "Voulez-vous vraiment quitter la partie ?"
- String `VICTORY_P1_MSG` = "Victoire !"
- String `VICTORY_P2_MSG` = "Defaite !"

Package Functions

- void `init` ()
- void `yesClicked` (`ActionEvent` event)
Validate end of game.
- void `displayShot` (`Shot` shot, int player)
Calls another function to draw the shot made according to the parameters.
- void `exitGame` (`ActionEvent` event)
Shows exit game message.
- void `noClicked` (`ActionEvent` event)
Cancel end of game.

Private Attributes

- `AnchorPane` `anchorPane`
- `Button` `exitButton`
- `Label` `gameState`
- `Label` `timerLabel`
- `Pane` `messageContainer`
- `Text` `messageTextContainer`
- `Button` `yesButton`
- `Button` `noButton`
- `GridPane` `table1`
- `GridPane` `table2`
- `AnchorPane` `chatPane`
- `AnchorPane` `chatPane1`
- `Label` `lblj1`
- `Label` `lblj2`
- boolean `waitExit` = false
- String `waitMsg`

Additional Inherited Members

7.55.1 Member Function Documentation

7.55.1.1 void guiTable.controllers.ObservationPhase.displayShot (*Shot shot*, int *player*) [package]

Calls another function to draw the shot made according to the parameters.

Parameters

<i>shot</i>	The shot to be added
<i>player</i>	Int that indicates the player

7.55.1.2 void guiTable.controllers.ObservationPhase.exitGame (*ActionEvent event*) [package]

Shows exit game message.

Parameters

<i>event</i>	
--------------	--

7.55.1.3 AnchorPane guiTable.controllers.ObservationPhase.getChatPane ()

7.55.1.4 String guiTable.controllers.ObservationPhase.getLblj1 ()

Returns

the lblj1

7.55.1.5 String guiTable.controllers.ObservationPhase.getLblj2 ()

Returns

the lblj2

7.55.1.6 void guiTable.controllers.ObservationPhase.init () [package]

7.55.1.7 void guiTable.controllers.ObservationPhase.logMsg (*String msg*)

log message into interface.

Parameters

<i>msg</i>	message to be displayed
------------	-------------------------

7.55.1.8 void guiTable.controllers.ObservationPhase.logYesNoMsg (String *msg*)

log yesNoMessage into interface.

Parameters

<i>msg</i>	message to be displayed
------------	-------------------------

7.55.1.9 void guiTable.controllers.ObservationPhase.noClicked (ActionEvent *event*) [package]

Cancel end of game.

7.55.1.10 void guiTable.controllers.ObservationPhase.placeShotTo (Shot *shot*, GridPane *grid*) [protected]

Adds a shot on a board.

Parameters

<i>shot</i>	The shot to be added
<i>grid</i>	The board where the shot will be added

7.55.1.11 void guiTable.controllers.ObservationPhase.setLblj1 (String *lblj1*)

Parameters

<i>lblj1</i>	the lblj1 to set
--------------	------------------

7.55.1.12 void guiTable.controllers.ObservationPhase.setLblj2 (String *lblj2*)

Parameters

<i>lblj2</i>	the lblj2 to set
--------------	------------------

7.55.1.13 void guiTable.controllers.ObservationPhase.setTableController (GuiTableController *c*)

Sets ths [GuiTableController](#).

Parameters

<i>c</i>	The GuiTableController
----------	--

7.55.1.14 void guiTable.controllers.ObservationPhase.setTurn (Boolean *t*)

Sets the turn of the players and and colors the boards.

Parameters

<i>t</i>	; Boolean that defines the turn. True means player 1, False means player 2
----------	--

7.55.1.15 void guiTable.controllers.ObservationPhase.showVictory (int *player*)

Shows victory message.

7.55.1.16 void guiTable.controllers.ObservationPhase.sunkABoat (GridPane *grid*, Boat *boat*) [protected]

Draws a sunked boat.

Parameters

<i>grid</i>	The board where the boat sunked
<i>boat</i>	The boat that has been sunked

7.55.1.17 void guiTable.controllers.ObservationPhase.sunkPlayerBoat (int *i*, Boat *boat*)

Calls another function to draw the sunked boat according to the parameters.

Parameters

<i>i</i>	Int that indicates the player
<i>boat</i>	The boat to be sunked

7.55.1.18 void guiTable.controllers.ObservationPhase.yesClicked (ActionEvent *event*) [package]

Validate end of game.

7.55.2 Member Data Documentation

7.55.2.1 AnchorPane guiTable.controllers.ObservationPhase.anchorPane [private]

7.55.2.2 AnchorPane guiTable.controllers.ObservationPhase.chatPane [private]

7.55.2.3 AnchorPane guiTable.controllers.ObservationPhase.chatPane1 [private]

- 7.55.2.4 String guiTable.controllers.ObservationPhase.EXIT_GAME_MSG = "Voulez-vous vraiment quitter la partie ?"
[protected]
- 7.55.2.5 Button guiTable.controllers.ObservationPhase.exitButton [private]
- 7.55.2.6 Label guiTable.controllers.ObservationPhase.gameState [private]
- 7.55.2.7 Label guiTable.controllers.ObservationPhase.lbj1 [private]
- 7.55.2.8 Label guiTable.controllers.ObservationPhase.lbj2 [private]
- 7.55.2.9 Pane guiTable.controllers.ObservationPhase.messageContainer [private]
- 7.55.2.10 Text guiTable.controllers.ObservationPhase.messageTextContainer [private]
- 7.55.2.11 String guiTable.controllers.ObservationPhase.MY_TURN_MSG = "A votre tour de jouer, cliquer sur une case puis sur le bouton valider" [protected]
- 7.55.2.12 Button guiTable.controllers.ObservationPhase.noButton [private]
- 7.55.2.13 String guiTable.controllers.ObservationPhase.OTHER_TURN_MSG = "Au tour de l'adversaire de jouer, merci de patienter" [protected]
- 7.55.2.14 final String guiTable.controllers.ObservationPhase.STYLE_MY_TURN = "-fx-background-color: #FFFFFF;"
[protected]
- 7.55.2.15 final String guiTable.controllers.ObservationPhase.STYLE_OTHER_TURN = "-fx-background-color: #EEEEEE;"
[protected]
- 7.55.2.16 GridPane guiTable.controllers.ObservationPhase.table1 [private]
- 7.55.2.17 GridPane guiTable.controllers.ObservationPhase.table2 [private]
- 7.55.2.18 GuiTableController guiTable.controllers.ObservationPhase.tableController [protected]
- 7.55.2.19 HashMap<Integer, GridPane> guiTable.controllers.ObservationPhase.tablePlayer
- 7.55.2.20 Label guiTable.controllers.ObservationPhase.timerLabel [private]
- 7.55.2.21 boolean guiTable.controllers.ObservationPhase.turn = false
- 7.55.2.22 String guiTable.controllers.ObservationPhase.VICTORY_P1_MSG = "Victoire !" [protected]
- 7.55.2.23 String guiTable.controllers.ObservationPhase.VICTORY_P2_MSG = "Defaite !" [protected]
- 7.55.2.24 boolean guiTable.controllers.ObservationPhase.waitExit = false [private]
- 7.55.2.25 String guiTable.controllers.ObservationPhase.waitMsg [private]
- 7.55.2.26 Button guiTable.controllers.ObservationPhase.yesButton [private]

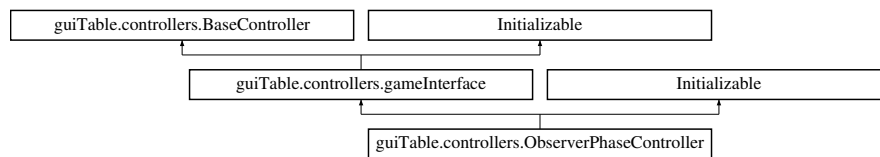
The documentation for this class was generated from the following file:

- src/main/java/guiTable/controllers/[ObservationPhase.java](#)

7.56 guiTable.controllers.ObserverPhaseController Class Reference

[ObserverPhaseController](#).

Inheritance diagram for guiTable.controllers.ObserverPhaseController:



Public Member Functions

- void [initialize](#) (URL location, ResourceBundle resources)
- void [setTurn](#) (Boolean turn)
Sets the turn of the players and and colors the boards.
- void [sunkPlayerBoat](#) (int player, [Boat](#) boat)

Package Functions

- void [displayShot](#) ([Shot](#) shot, int player)
- void [exitGame](#) (ActionEvent event)
- void [noClicked](#) (ActionEvent event)
Cancel end of game.

Private Attributes

- GridPane [table1](#)
- GridPane [table2](#)
- Label [gameState](#)
- HashMap< Integer, GridPane > [tablePlayer](#)

Additional Inherited Members

7.56.1 Detailed Description

[ObserverPhaseController](#).

7.56.2 Member Function Documentation

7.56.2.1 void `guiTable.controllers.ObserverPhaseController.displayShot (Shot shot, int player)` [package]

7.56.2.2 void `guiTable.controllers.ObserverPhaseController.exitGame (ActionEvent event)` [package]

7.56.2.3 void `guiTable.controllers.ObserverPhaseController.initialize (URL location, ResourceBundle resources)`

7.56.2.4 void `guiTable.controllers.ObserverPhaseController.noClicked (ActionEvent event)` [package]

Cancel end of game.

7.56.2.5 void `guiTable.controllers.ObserverPhaseController.setTurn (Boolean turn)`

Sets the turn of the players and and colors the boards.

Parameters

<i>turn</i>	; Boolean that defines the turn. True means player 1, False means player 2
-------------	--

7.56.2.6 void guiTable.controllers.ObserverPhaseController.sunkPlayerBoat (int *player*, Boat *boat*)

7.56.3 Member Data Documentation

7.56.3.1 Label guiTable.controllers.ObserverPhaseController.gameState [private]

7.56.3.2 GridPane guiTable.controllers.ObserverPhaseController.table1 [private]

7.56.3.3 GridPane guiTable.controllers.ObserverPhaseController.table2 [private]

7.56.3.4 HashMap<Integer, GridPane> guiTable.controllers.ObserverPhaseController.tablePlayer [private]

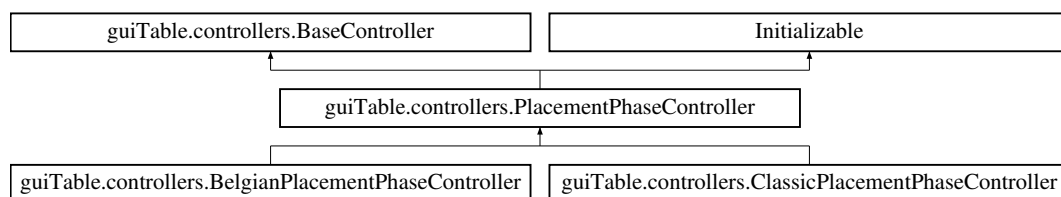
The documentation for this class was generated from the following file:

- [src/main/java/guiTable/controllers/ObserverPhaseController.java](#)

7.57 guiTable.controllers.PlacementPhaseController Class Reference

[PlacementPhaseController](#).

Inheritance diagram for guiTable.controllers.PlacementPhaseController:



Public Member Functions

- void [logMsg](#) (String msg, String...param)
log message into interface.
- void [initialize](#) (URL location, ResourceBundle resources)
The method initialize starts the window and assigns values [BoatDrawing](#) objects and methods to the window's objects.
- void [setPlacementTime](#) (Integer placementTime)
Sets the maximum time of placement of boats.
- boolean [isIsValidate](#) ()
- void [setIsValidate](#) (boolean isValidate)
- AnchorPane [getChatPane](#) ()
Returns the chat pane.

Protected Member Functions

- void [closeMsg](#) ()
close message when click on it
- void [initBoatMap](#) ()
method to put boat in the boatMap.
- void [onValidate](#) ()
Trigger validation of placement phase.
- boolean [allBoatsArePlaced](#) ()
Check if all boats are placed (ie.
- EventHandler< MouseEvent > [activateBoat](#) ()
Activates the boat when the user clicks on it.
- EventHandler< MouseEvent > [enableRotation](#) ()
Enables the boat rotation, this only happens when the mouse is over the grid.
- EventHandler< MouseEvent > [disableRotation](#) ()
Disables the boat rotation when the mouse exits the grid.
- EventHandler< MouseEvent > [drawBoatsNewPosition](#) ()
Moves the boat on the grid when the mouse moves over the squares.
- void [draw](#) ([BoatDrawing](#) boat, Integer collIndex, Integer rowIndex)
Draws and updates the boat's position.
- EventHandler< KeyEvent > [playKeyEvent](#) ()
Rotates the active boat when the user press R and deletes the active boat when the user presses delete.
- void [drawRotation](#) ([BoatDrawing](#) boat)
Rotates the boat with 90 ° and updates the view.
- void [reinitBoat](#) ([BoatDrawing](#) boat)
Relocates the boat in the initial position and deactivates it.
- boolean [positionCorrect](#) ([BoatDrawing](#) activeBoat)
Checks if the user cans put the boat at the selected position.
- EventHandler< MouseEvent > [MousePlaceBoat](#) ()
Unactivates the boat when it is placed over the grid.
- void [placeBoat](#) ([BoatDrawing](#) myBoat)
Tries to place a boat on the board.
- void [desactiveBoat](#) ()
Desactivates the active boat.
- void [timelsOver](#) ()
Places the boats randomly if time's over and there are boats to place.

Protected Attributes

- Timeline [timeline](#)
- Label [timerLabel](#)
- LocalTime [time](#)
- boolean [rotationIsValide](#)
- [BoatDrawing](#) [activeBoat](#)
- HashMap< Rectangle, [BoatDrawing](#) > [boatMap](#)

Static Protected Attributes

- static final int [RANDOM_ROTATION](#) = 2

Private Attributes

- AnchorPane [anchorPane](#)
- GridPane [table](#)
- Button [valider](#)
- AnchorPane [chatPane](#)
- Text [messageContainer](#)
- final String [EXPLAIN_PLACEMENT](#) = "press R to rotate [Boat](#) and DEL to reinitialize boat"
- LocalTime [timePlacement](#)
- boolean [isValidate](#) = false

Additional Inherited Members

7.57.1 Detailed Description

[PlacementPhaseController](#).

7.57.2 Member Function Documentation

7.57.2.1 `EventHandler<MouseEvent> guiTable.controllers.PlacementPhaseController.activateBoat ()` `[protected]`

Activates the boat when the user clicks on it.

Returns

mousePressHandler The handler of the event (Click over the boat).

7.57.2.2 `boolean guiTable.controllers.PlacementPhaseController.allBoatsArePlaced ()` `[protected]`

Check if all boats are placed (ie.

every boats are on the gird)

Returns

boolean true if all boats are placed, false else

7.57.2.3 `void guiTable.controllers.PlacementPhaseController.closeMsg ()` `[protected]`

close message when click on it

7.57.2.4 `void guiTable.controllers.PlacementPhaseController.desactiveBoat ()` `[protected]`

Desactivates the active boat.

7.57.2.5 EventHandler<MouseEvent> guiTable.controllers.PlacementPhaseController.disableRotation () [protected]

Disables the boat rotation when the mouse exits the grid.

Returns

mouseLocationHandler The handler of the event (Mouse exits the grid).

7.57.2.6 void guiTable.controllers.PlacementPhaseController.draw (BoatDrawing *boat*, Integer *colIndex*, Integer *rowIndex*)
[protected]

Draws and updates the boat's position.

Parameters

<i>boat</i>	The boat that will be drawn.
<i>colIndex</i>	The grid column of its position.
<i>rowIndex</i>	The grid row of tis position.

7.57.2.7 `EventHandler<MouseEvent> guiTable.controllers.PlacementPhaseController.drawBoatsNewPosition ()` [protected]

Moves the boat on the grid when the mouse moves over the squares.

Returns

mousePositionHandler The handler of the event (Mouse enters a grid position).

7.57.2.8 `void guiTable.controllers.PlacementPhaseController.drawRotation (BoatDrawing boat)` [protected]

Rotates the boat with 90° and updates the view.

Parameters

<i>boat</i>	The boat that will be rotated
-------------	-------------------------------

7.57.2.9 `EventHandler<MouseEvent> guiTable.controllers.PlacementPhaseController.enableRotation ()` [protected]

Enables the boat rotation, this only happens when the mouse is over the grid.

Returns

mouseLocationHandler The handler of the event (Mouse enters ther grid).

7.57.2.10 `AnchorPane guiTable.controllers.PlacementPhaseController.getChatPane ()`

Returns the chat pane.

Returns

The chat pane

7.57.2.11 `void guiTable.controllers.PlacementPhaseController.initBoatMap ()` [protected]

method to put boat in the boatMap.

method must be override by subclasses.

7.57.2.12 `void guiTable.controllers.PlacementPhaseController.initialize (URL location, ResourceBundle resources)`

The method initialize starts the window and assigns values [BoatDrawing](#) objects and methods to the window's objects.

Parameters

<i>location</i>	
<i>resources</i>	

7.57.2.13 `boolean guiTable.controllers.PlacementPhaseController.isIsValidate ()`

Returns

the isValidate

7.57.2.14 `void guiTable.controllers.PlacementPhaseController.logMsg (String msg, String... param)`

log message into interface.

Parameters

<i>msg</i>	message to be displayed
<i>param</i>	list of optionnal parameter, all strings of param are display in CLI

7.57.2.15 `EventHandler<MouseEvent> guiTable.controllers.PlacementPhaseController.MousePlaceBoat ()`
[protected]

Unactivates the boat when it is placed over the grid.

Returns

mousePressGridHandler The handler of the event (Click over the grid).

7.57.2.16 `void guiTable.controllers.PlacementPhaseController.onValidate ()` [protected]

Trigger validation of placement phase.

7.57.2.17 `void guiTable.controllers.PlacementPhaseController.placeBoat (BoatDrawing myBoat)` [protected]

Tries to place a boat on the board.

Parameters

<i>myBoat</i>	to boat to be placed
---------------	----------------------

7.57.2.18 `EventHandler<KeyEvent> guiTable.controllers.PlacementPhaseController.playKeyEvent ()` [protected]

Rotates the active boat when the user press R and deletes the active boat when the user presses delete.

Returns

keyEeventHandler The handler of the event (Users types any key).

7.57.2.19 `boolean guiTable.controllers.PlacementPhaseController.positionCorrect (BoatDrawing activeBoat)`
[protected]

Checks if the user cans put the boat at the selected position.

Parameters

<i>activeBoat</i>	The active boat that will have its positions checked.
-------------------	---

Returns

True if the position is correct.

7.57.2.20 `void guiTable.controllers.PlacementPhaseController.reinitBoat (BoatDrawing boat)` [protected]

Relocates the boat in the initial position and deactivates it.

Parameters

<i>boat</i>	The boat that will be relocated and deactivated.
-------------	--

7.57.2.21 `void guiTable.controllers.PlacementPhaseController.setIsValidate (boolean isValid)`

Parameters

<i>isValid</i>	the isValid to set
----------------	--------------------

7.57.2.22 `void guiTable.controllers.PlacementPhaseController.setPlacementTime (Integer placementTime)`

Sets the maximum time of placement of boats.

Parameters

<i>placementTime</i>	The time of placement
----------------------	-----------------------

7.57.2.23 void guiTable.controllers.PlacementPhaseController.timelsOver () [protected]

Places the boats randomly if time's over and there are boats to place.

7.57.3 Member Data Documentation

7.57.3.1 BoatDrawing guiTable.controllers.PlacementPhaseController.activeBoat [protected]

7.57.3.2 AnchorPane guiTable.controllers.PlacementPhaseController.anchorPane [private]

7.57.3.3 HashMap<Rectangle, BoatDrawing> guiTable.controllers.PlacementPhaseController.boatMap [protected]

7.57.3.4 AnchorPane guiTable.controllers.PlacementPhaseController.chatPane [private]

7.57.3.5 final String guiTable.controllers.PlacementPhaseController.EXPLAIN_PLACEMENT = "press R to rotate Boat and DEL to reinitialize boat" [private]

7.57.3.6 boolean guiTable.controllers.PlacementPhaseController.isValidate = false [private]

7.57.3.7 Text guiTable.controllers.PlacementPhaseController.messageContainer [private]

7.57.3.8 final int guiTable.controllers.PlacementPhaseController.RANDOM_ROTATION = 2 [static], [protected]

7.57.3.9 boolean guiTable.controllers.PlacementPhaseController.rotationIsValide [protected]

7.57.3.10 GridPane guiTable.controllers.PlacementPhaseController.table [private]

7.57.3.11 LocalTime guiTable.controllers.PlacementPhaseController.time [protected]

7.57.3.12 Timeline guiTable.controllers.PlacementPhaseController.timeline [protected]

7.57.3.13 LocalTime guiTable.controllers.PlacementPhaseController.timePlacement [private]

7.57.3.14 Label guiTable.controllers.PlacementPhaseController.timerLabel [protected]

7.57.3.15 Button guiTable.controllers.PlacementPhaseController.valider [private]

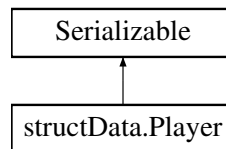
The documentation for this class was generated from the following file:

- src/main/java/guiTable/controllers/[PlacementPhaseController.java](#)

7.58 structData.Player Class Reference

[Player](#) is a class for players in a game.

Inheritance diagram for structData.Player:



Public Member Functions

- [Player](#) ([DataUser](#) dUser)
Constructor by default.
- [Player](#) ([Profile](#) p)
Constructor for a [Player](#).
- boolean [isReady](#) ()
Accessor for ready.
- void [setReady](#) (boolean value)
Mutator for ready.
- [Player](#) ([Profile](#) p, List< [Boat](#) > listBoatsData, HashSet< [Shot](#) > listShotsData)
Constructor with parameters.
- [Profile](#) [getProfile](#) ()
Accessors for [Profile](#).
- List< [Boat](#) > [getListBoats](#) ()
Accessors for listBoats.
- HashSet< [Shot](#) > [getListShots](#) ()
Accessors for listShots.
- void [setProfile](#) ([Profile](#) p)
Mutator for profile.
- void [setListBoats](#) (List< [Boat](#) > listBoatsData)
Mutator for listBoats.
- void [setListShots](#) (HashSet< [Shot](#) > listShotsData)
Mutator for listShots.
- [Player](#) [clonePlayer](#) ([Player](#) playerClone)
Clone function to copy a [Player](#).
- void [addBoat](#) ([Boat](#) newBoat)
Add a new boat to the list of boats.
- void [addShot](#) ([Shot](#) newShot)
Add a new shot to the list of shots.
- Boolean [compareProfileToPlayer](#) ([Profile](#) p)
Function to compare a profile with the current player's profile.

Static Package Attributes

- static final long [serialVersionUID](#) = 5L

Private Attributes

- [Profile](#) `profile`
- `List< Boat >` `listBoats`
- `HashSet< Shot >` `listShots`
- `boolean` `ready`

7.58.1 Detailed Description

[Player](#) is a class for players in a game.

7.58.2 Constructor & Destructor Documentation

7.58.2.1 structData.Player.Player ([DataUser](#) *dUser*)

Constructor by default.

Parameters

<i>dUser</i>	: the parent class DataUser of the profile of the new player
--------------	--

7.58.2.2 structData.Player.Player ([Profile](#) *p*)

Constructor for a [Player](#).

Parameters

<i>dUser</i>	: the parent class DataUser of the profil of the new player
--------------	---

7.58.2.3 structData.Player.Player ([Profile](#) *p*, `List< Boat >` *listBoatsData*, `HashSet< Shot >` *listShotsData*)

Constructor with parameters.

Parameters

<i>p</i>	: the profile of the new player
<i>listBoatsData</i>	: the list of boats of the new player
<i>listShotsData</i>	: the list of shots of the new player

7.58.3 Member Function Documentation

7.58.3.1 void structData.Player.addBoat ([Boat](#) *newBoat*)

Add a new boat to the list of boats.

Parameters

<i>newBoat</i>	: the new boat to be added into the list of boats
----------------	---

7.58.3.2 void structData.Player.addShot (Shot *newShot*)

Add a new shot to the list of shots.

Parameters

<i>newShot</i>	: the new shot to be added into the list of boats
----------------	---

7.58.3.3 Player structData.Player.clonePlayer (Player *playerClone*)

Clone function to copy a [Player](#).

Parameters

<i>playerClone</i>	: the player to be cloned
--------------------	---------------------------

Returns

the player who has called this method to clone all the information of the *playerClone*

7.58.3.4 Boolean structData.Player.compareProfileToPlayer (Profile *p*)

Function to compare a profile with the current player's profile.

Parameters

<i>p</i>	profile to compare with the current player's profile
----------	--

Returns

a boolean wich is true if profiles are equals

7.58.3.5 List<Boat> structData.Player.getListBoats ()

Accessors for listBoats.

Returns

the list of boats of the player

7.58.3.6 `HashSet<Shot> structData.Player.getListShots ()`

Accessors for listShots.

Returns

the list of shots of the player

7.58.3.7 `Profile structData.Player.getProfile ()`

Accessors for [Profile](#).

Returns

the profile of the player

7.58.3.8 `boolean structData.Player.isReady ()`

Accessor for ready.

Returns

a boolean that is true if the player is ready to play

7.58.3.9 `void structData.Player.setListBoats (List< Boat > listBoatsData)`

Mutator for listBoats.

Parameters

<i>listBoatsData</i>	: the new value of the list of boats of the player
----------------------	--

7.58.3.10 `void structData.Player.setListShots (HashSet< Shot > listShotsData)`

Mutator for listShots.

Parameters

<i>listShotsData</i>	: the new value of the list of shots of the player
----------------------	--

7.58.3.11 `void structData.Player.setProfile (Profile p)`

Mutator for profile.

Parameters

<i>p</i>	: the new value of the profile of the player
----------	--

7.58.3.12 void structData.Player.setReady (boolean *value*)

Mutator for ready.

Parameters

<i>value</i>	: the ready value
--------------	-------------------

7.58.4 Member Data Documentation

7.58.4.1 List<Boat> structData.Player.listBoats [private]

7.58.4.2 HashSet<Shot> structData.Player.listShots [private]

7.58.4.3 Profile structData.Player.profile [private]

7.58.4.4 boolean structData.Player.ready [private]

7.58.4.5 final long structData.Player.serialVersionUID = 5L [static], [package]

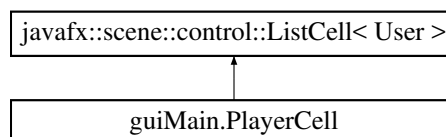
The documentation for this class was generated from the following file:

- src/main/java/structData/[Player.java](#)

7.59 guiMain.PlayerCell Class Reference

[PlayerCell](#), descendant class ListCell<User>, display the user name and offers the possibility watch his profil.

Inheritance diagram for guiMain.PlayerCell:



Public Member Functions

- [PlayerCell](#) (menuController c)
Init the gui parameter of the cell.

Protected Member Functions

- void `updateItem` (final `User` `user`, boolean `empty`)
Update the cell.

Package Attributes

- `HBox` `hbox` = new `HBox()`
- `HBox` `lookButtonBox` = new `HBox()`
- `Label` `name` = new `Label()`
- `Button` `lookButton` = new `Button("")`
- `menuController` `controller`

7.59.1 Detailed Description

`PlayerCell`, descendant class `ListCell<User>`, display the user name and offers the possibility watch his profil.

Author

IHM-MAIN Module

7.59.2 Constructor & Destructor Documentation

7.59.2.1 `guiMain.PlayerCell.PlayerCell (menuController c)`

Init the gui parameter of the cell.

Parameters

<code>c</code>	: controller necessary for futur update.
----------------	--

7.59.3 Member Function Documentation

7.59.3.1 `void guiMain.PlayerCell.updateItem (final User user, boolean empty)` `[protected]`

Update the cell.

Parameters

<code>user</code>	: user's cell to update.
-------------------	--------------------------

7.59.4 Member Data Documentation

7.59.4.1 `menuController guiMain.PlayerCell.controller` `[package]`

7.59.4.2 `HBox guiMain.PlayerCell.hbox = new HBox()` [package]

7.59.4.3 `Button guiMain.PlayerCell.lookButton = new Button("")` [package]

7.59.4.4 `HBox guiMain.PlayerCell.lookButtonBox = new HBox()` [package]

7.59.4.5 `Label guiMain.PlayerCell.name = new Label()` [package]

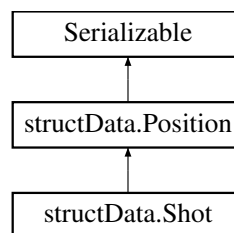
The documentation for this class was generated from the following file:

- `src/main/java/guiMain/PlayerCell.java`

7.60 structData.Position Class Reference

[Position](#) is a class of a position in a table.

Inheritance diagram for structData.Position:



Public Member Functions

- [Position](#) ()
Position : default constructor.
- [Position](#) (int newX, int newY, Boolean newTouched)
Position : constructor with all parameters.
- [Position](#) ([Position](#) p)
Position : constructor with a [Position](#).
- [Position](#) clone ([Position](#) p)
Method to copy a [Position](#).
- void [setX](#) (int x)
Mutator for x.
- int [getX](#) ()
Accessor for x.
- void [setY](#) (int y)
Mutator for y.
- int [getY](#) ()
Accessor for y.
- void [setTouched](#) (Boolean touched)
Mutator for touched.
- Boolean [getTouched](#) ()
Accessor for touched.

Protected Attributes

- int [x](#)
- int [y](#)
- Boolean [touched](#)

Static Package Attributes

- static final long [serialVersionUID](#) = 6L

7.60.1 Detailed Description

[Position](#) is a class of a position in a table.

7.60.2 Constructor & Destructor Documentation

7.60.2.1 `structData.Position.Position ()`

[Position](#) : default constructor.

7.60.2.2 `structData.Position.Position (int newX, int newY, Boolean newTouched)`

[Position](#) : constructor with all parameters.

Parameters

<i>newX</i>	x coordinate
<i>newY</i>	y coordinate
<i>newTouched</i>	boolean equal to 1 if the position is touched

7.60.2.3 `structData.Position.Position (Position p)`

[Position](#) : constructor with a [Position](#).

Parameters

<i>p</i>	a Position
----------	----------------------------

7.60.3 Member Function Documentation

7.60.3.1 `Position structData.Position.clone (Position p)`

Method to copy a [Position](#).

Parameters

<i>p</i>	a Position to copy
----------	------------------------------------

Returns

a copied [Position](#)

7.60.3.2 Boolean structData.Position.getTouched ()

Accessor for touched.

Returns

touched : boolean indicating if the position is touched or not

7.60.3.3 int structData.Position.getX ()

Accessor for x.

Returns

the position's x as a int

7.60.3.4 int structData.Position.getY ()

Accessor for y.

Returns

the position's y as a int

7.60.3.5 void structData.Position.setTouched (Boolean *touched*)

Mutator for touched.

Parameters

<i>touched</i>	boolean indicating if the position is touched or not
----------------	--

7.60.3.6 void structData.Position.setX (int *x*)

Mutator for x.

Parameters

<code>x</code>	the position's x
----------------	------------------

7.60.3.7 void structData.Position.setY (int y)

Mutator for y.

Parameters

<code>y</code>	the position's y
----------------	------------------

7.60.4 Member Data Documentation

7.60.4.1 final long structData.Position.serialVersionUID = 6L [static], [package]

7.60.4.2 Boolean structData.Position.touched [protected]

7.60.4.3 int structData.Position.x [protected]

7.60.4.4 int structData.Position.y [protected]

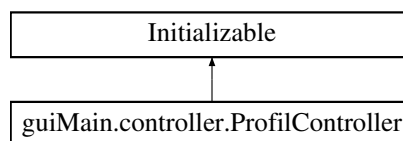
The documentation for this class was generated from the following file:

- src/main/java/structData/[Position.java](#)

7.61 guiMain.controller.ProfilController Class Reference

[GameCell](#), implement interface Initializable, display the user profil.

Inheritance diagram for guiMain.controller.ProfilController:



Public Member Functions

- void [initialize](#) (URL location, ResourceBundle resources)
Call at the initialization of the component.
- void [init](#) ([User](#) user)
Call the data module to recuperate the user informations.
- void [setProfil](#) ([Profile](#) profile)
Set user profile data into corresponding champs.
- void [setMainController](#) ([GuiMainController](#) c)
Set [GuiMainController](#) mainController.

Package Functions

- void [backToTheMenu](#) (ActionEvent event)
Return to menu window.

Private Attributes

- [GuiMainController](#) [mainController](#)
- Label [nameTitle](#)
- Label [userName](#)
- Label [firstName](#)
- Label [lastName](#)
- Label [birthdate](#)
- Label [numberOfGame](#)
- Label [numberOfGameWon](#)
- Label [numberOfGameLost](#)
- Button [backButton](#)
- ImageView [userAvatar](#)

7.61.1 Detailed Description

[GameCell](#), implement interface Initializable, display the user profil.

Author

IHM-MAIN Module

7.61.2 Member Function Documentation

7.61.2.1 void [guiMain.controller.ProfilController.backToTheMenu](#) ([ActionEvent](#) *event*) [[package](#)]

Return to menu window.

Parameters

<i>event</i>	: backButton event
--------------	------------------------------------

7.61.2.2 void [guiMain.controller.ProfilController.init](#) ([User](#) *user*)

Call the data module to recuperate the user informations.

Parameters

<i>user</i>	: user that we want to recuperate the informations
-------------	--

7.61.2.3 void guiMain.controller.ProfilController.initialize (URL *location*, ResourceBundle *resources*)

Call at the initialization of the component.

Parameters

<i>location</i>	↔
:	:
<i>resources</i>	↔
:	:

7.61.2.4 void guiMain.controller.ProfilController.setMainController (GuiMainController *c*)

Set [GuiMainController](#) mainController.

Parameters

<i>c</i>	: GuiMainController
----------	-------------------------------------

7.61.2.5 void guiMain.controller.ProfilController.setProfil (Profile *profile*)

Set user profile data into corresponding champs.

Parameters

<i>profile</i>	the user profile that should be display
----------------	---

7.61.3 Member Data Documentation

7.61.3.1 Button guiMain.controller.ProfilController.backButton [private]

7.61.3.2 Label guiMain.controller.ProfilController.birthdate [private]

7.61.3.3 Label guiMain.controller.ProfilController.firstName [private]

7.61.3.4 Label guiMain.controller.ProfilController.lastName [private]

7.61.3.5 GuiMainController guiMain.controller.ProfilController.mainController [private]

7.61.3.6 Label guiMain.controller.ProfilController.nameTitle [private]

7.61.3.7 Label guiMain.controller.ProfilController.numberOfGame [private]

7.61.3.8 Label guiMain.controller.ProfilController.numberOfGameLost [private]

7.61.3.9 Label `guiMain.controller.ProfilController.numberOfGameWon` [private]

7.61.3.10 ImageView `guiMain.controller.ProfilController.userAvatar` [private]

7.61.3.11 Label `guiMain.controller.ProfilController.userName` [private]

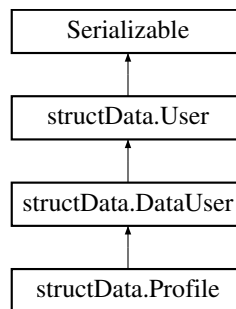
The documentation for this class was generated from the following file:

- `src/main/java/guiMain/controller/ProfilController.java`

7.62 structData.Profile Class Reference

[Profile](#) is a class for the user's profile.

Inheritance diagram for `structData.Profile`:



Public Member Functions

- [Profile](#) ([DataUser](#) dUser)
Constructor with only a [DataUser](#).
- [Profile](#) ([DataUser](#) dUser, String pathToImage, String newLastname, String newName, Date newBirthdate)
Constructor with all paramters.
- [Profile](#) ([DataUser](#) dUser, ImageIcon image, String newLastname, String newName, Date newBirthdate)
Constructor with all paramters.
- [Profile clone](#) ([Profile](#) p)
Method to copy a [Profile](#).
- void [setAvatar](#) (ImageIcon i)
Mutator for the avatar.
- void [setAvatar](#) (String path)
Mutator for the avatar.
- ImageIcon [getAvatar](#) ()
Accessor for the avatar.
- Image [getImage](#) ()
Accessor for the image from avatar.
- void [setLastname](#) (String ln)
Mutator for the lastname.
- String [getLastname](#) ()
Accessor for the lastname.

- void [setName](#) (String n)
Mutator for the name.
- String [getName](#) ()
Accessor for the name.
- void [setBirthdate](#) (Date b)
Mutator for the birthdate.
- Date [getBirthdate](#) ()
Accessor for the birthdate.
- void [setGamesPlayed](#) (int nb)
Mutator for gamesPlayed.
- int [getGamesPlayed](#) ()
Accessor for gamesPlayed.
- void [setGamesWon](#) (int nb)
Mutator for gamesWon.
- int [getGamesWon](#) ()
Accessor for gamesWon.
- void [setGamesLost](#) (int nb)
Mutator for gamesLost.
- int [getGamesLost](#) ()
Accessor for gamesLost.
- void [setGamesAborted](#) (int nb)
Mutator for gamesAborted.
- int [getGamesAborted](#) ()
Accessor for gamesAborted.
- void [saveProfile](#) ()
Save new profile in a local file.
- boolean [cleanUp](#) (File file) throws IOException
Safe clean of file.
- void [saveeditedProfile](#) ()
Save edited profil.

Private Attributes

- ImageIcon [avatar](#)
- String [lastname](#)
- String [name](#)
- Date [birthdate](#)
- int [gamesPlayed](#)
- int [gamesWon](#)
- int [gamesLost](#)
- int [gamesAborted](#)

Additional Inherited Members

7.62.1 Detailed Description

[Profile](#) is a class for the user's profile.

7.62.2 Constructor & Destructor Documentation

7.62.2.1 structData.Profile.Profile ([DataUser dUser](#))

Constructor with only a [DataUser](#).

Parameters

<i>dUser</i>	a DataUser
--------------	----------------------------

7.62.2.2 `structData.Profile.Profile (DataUser dUser, String pathToImage, String newLastname, String newName, Date newBirthdate)`

Constructor with all paramters.

Parameters

<i>dUser</i>	a DataUser
<i>pathToImage</i>	path to the image for avatar
<i>newLastname</i>	a lastname
<i>newName</i>	a name
<i>newBirthdate</i>	a birthdate

7.62.2.3 `structData.Profile.Profile (DataUser dUser, ImageIcon image, String newLastname, String newName, Date newBirthdate)`

Constructor with all paramters.

Parameters

<i>dUser</i>	a DataUser
<i>image</i>	image for avatar
<i>newLastname</i>	a lastname
<i>newName</i>	a name
<i>newBirthdate</i>	a birthdate

7.62.3 Member Function Documentation

7.62.3.1 `boolean structData.Profile.cleanUp (File file)` throws `IOException`

Safe clean of file.

Parameters

<i>file</i>	to delete
-------------	-----------

7.62.3.2 `Profile structData.Profile.clone (Profile p)`

Method to copy a [Profile](#).

Parameters

p	a Profile to copy
-----	-----------------------------------

Returns

a copied [Profile](#)

7.62.3.3 `ImageIcon structData.Profile.getAvatar ()`

Accessor for the avatar.

Returns

an avatar as an image

7.62.3.4 `Date structData.Profile.getBirthdate ()`

Accessor for the birthdate.

Returns

a birthdate as a date

7.62.3.5 `int structData.Profile.getGamesAborted ()`

Accessor for gamesAborted.

Returns

the number of games aborted as an integer

7.62.3.6 `int structData.Profile.getGamesLost ()`

Accessor for gamesLost.

Returns

the number of games lost as an integer

7.62.3.7 `int structData.Profile.getGamesPlayed ()`

Accessor for gamesPlayed.

Returns

the number of games played as an integer

7.62.3.8 int structData.Profile.getGamesWon ()

Accessor for gamesWon.

Returns

the number of games won as an integer

7.62.3.9 Image structData.Profile.getImage ()

Accessor for the image from avatar.

Returns

an avatar as an image

7.62.3.10 String structData.Profile.getLastname ()

Accessor for the lastname.

Returns

a lastname as a string

7.62.3.11 String structData.Profile.getName ()

Accessor for the name.

Returns

a name as a string

7.62.3.12 void structData.Profile.saveeditedProfile ()

Save edited profil.

7.62.3.13 void structData.Profile.saveProfile ()

Save new profile in a local file.

7.62.3.14 void structData.Profile.setAvatar (ImageIcon i)

Mutator for the avatar.

Parameters

<i>i</i>	an image
----------	----------

7.62.3.15 void structData.Profile.setAvatar (String *path*)

Mutator for the avatar.

Parameters

<i>path</i>	path to image
-------------	---------------

7.62.3.16 void structData.Profile.setBirthdate (Date *b*)

Mutator for the birthdate.

Parameters

<i>b</i>	a birthdate
----------	-------------

7.62.3.17 void structData.Profile.setGamesAborted (int *nb*)

Mutator for gamesAborted.

Parameters

<i>nb</i>	the number of games aborted as an integer
-----------	---

7.62.3.18 void structData.Profile.setGamesLost (int *nb*)

Mutator for gamesLost.

Parameters

<i>nb</i>	the number of games lost as an integer
-----------	--

7.62.3.19 void structData.Profile.setGamesPlayed (int *nb*)

Mutator for gamesPlayed.

Parameters

<i>nb</i>	the number of games played
-----------	----------------------------

7.62.3.20 void structData.Profile.setGamesWon (int *nb*)

Mutator for gamesWon.

Parameters

<i>nb</i>	the number of games won as an integer
-----------	---------------------------------------

7.62.3.21 void structData.Profile.setLastname (String *ln*)

Mutator for the lastname.

Parameters

<i>ln</i>	a lastname
-----------	------------

7.62.3.22 void structData.Profile.setName (String *n*)

Mutator for the name.

Parameters

<i>n</i>	a name
----------	--------

7.62.4 Member Data Documentation

7.62.4.1 ImageIcon structData.Profile.avatar [private]

7.62.4.2 Date structData.Profile.birthdate [private]

7.62.4.3 int structData.Profile.gamesAborted [private]

7.62.4.4 int structData.Profile.gamesLost [private]

7.62.4.5 int structData.Profile.gamesPlayed [private]

7.62.4.6 int structData.Profile.gamesWon [private]

7.62.4.7 String structData.Profile.lastname [private]

7.62.4.8 String structData.Profile.name [private]

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

- src/main/java/structData/[Profile.java](#)

7.63 guiTable.controllers.ProfileController Class Reference

Public Member Functions

- void [init](#) ()

7.63.1 Member Function Documentation

7.63.1.1 void guiTable.controllers.ProfileController.init ()

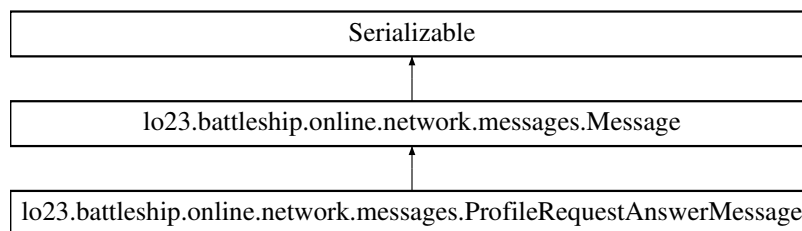
The documentation for this class was generated from the following file:

- [src/main/java/guiTable/controllers/ProfileController.java](#)

7.64 lo23.battleship.online.network.messages.ProfileRequestAnswerMessage Class Reference

Network message class, extends [Message](#) class.

Inheritance diagram for lo23.battleship.online.network.messages.ProfileRequestAnswerMessage:



Public Member Functions

- String [getType](#) ()
Returns the type of the message.
- void [process](#) ([IDataCom](#) IData, [InetAddress](#) senderAddress)
Notifies the data package interface that profile which the local user asked for, has been received.

Package Functions

- [ProfileRequestAnswerMessage](#) ([Profile](#) profile)
Allocates a new.

Private Attributes

- [User](#) user
- [Profile](#) profile

Additional Inherited Members

7.64.1 Detailed Description

Network message class, extends [Message](#) class.

[Message](#) sent to provide the Profile requested by the network.

Author

COM Module

7.64.2 Constructor & Destructor Documentation

7.64.2.1 `Io23.battleship.online.network.messages.ProfileRequestAnswerMessage.ProfileRequestAnswerMessage (Profile profile) [package]`

Allocates a new.

[ProfileRequestAnswerMessage](#)

object

Parameters

<i>profile</i>	:
	Profile
	the Profile class which needs to be sent.

7.64.3 Member Function Documentation

7.64.3.1 `String Io23.battleship.online.network.messages.ProfileRequestAnswerMessage.getType ()`

Returns the type of the message.

Implementation of an abstract method.

Returns

type :
`String`

7.64.3.2 `void Io23.battleship.online.network.messages.ProfileRequestAnswerMessage.process (IDataCom IData, InetAddress senderAddress)`

Notifies the data package interface that profile which the local user asked for, has been received.

Parameters

<i>IData</i>	: IDataCom instance of IDataCom interface.
<i>senderAddress</i>	: InetAddress sender's IP address

7.64.4 Member Data Documentation

7.64.4.1 **Profile** lo23.battleship.online.network.messages.ProfileRequestAnswerMessage.profile [private]

7.64.4.2 **User** lo23.battleship.online.network.messages.ProfileRequestAnswerMessage.user [private]

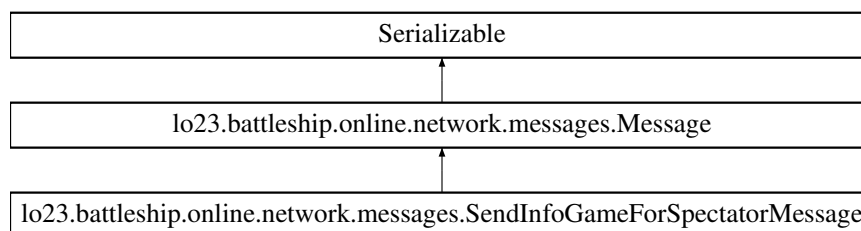
The documentation for this class was generated from the following file:

- src/main/java/lo23/battleship/online/network/messages/[ProfileRequestAnswerMessage.java](#)

7.65 lo23.battleship.online.network.messages.SendInfoGameForSpectatorMessage Class Reference

This class implements the message sent when the local user has been accepted as spectator.

Inheritance diagram for lo23.battleship.online.network.messages.SendInfoGameForSpectatorMessage:



Public Member Functions

- [SendInfoGameForSpectatorMessage](#) ([Game](#) g, [User](#) s)
- String [getType](#) ()
Returns the type of the message.
- void [process](#) ([IDataCom](#) IData, [InetAddress](#) senderAddress)
Sets the initial state of the game the spectator(local user) is watching using the data package interface IData

Package Attributes

- [Game game](#)
- [User spec](#)

7.65.1 Detailed Description

This class implements the message sent when the local user has been accepted as spectator.

The local user receives this message with the current state(shots and sunk boats etc. ...) of the game.

See also

[Message](#)

Author

COM Module

7.65.2 Constructor & Destructor Documentation

7.65.2.1 `Io23.battleship.online.network.messages.SendInfoGameForSpectatorMessage.SendInfoGameForSpectatorMessage (Game g, User s)`

7.65.3 Member Function Documentation

7.65.3.1 `String Io23.battleship.online.network.messages.SendInfoGameForSpectatorMessage.getType ()`

Returns the type of the message.

Implementation of an abstract method.

Returns

type :

`String`

7.65.3.2 `void Io23.battleship.online.network.messages.SendInfoGameForSpectatorMessage.process (IDataCom IData, InetAddress senderAddress)`

Sets the initial state of the game the spectator(local user) is watching using the data package interface `IData`

Parameters

<i>IData</i>	: <code>IDataCom</code> instance of <code>IDataCom</code> interface.
<i>senderAddress</i>	: <code>InetAddress</code> sender's IP address

7.65.4 Member Data Documentation

7.65.4.1 **Game** `lo23.battleship.online.network.messages.SendInfoGameForSpectatorMessage.game` [package]

7.65.4.2 **User** `lo23.battleship.online.network.messages.SendInfoGameForSpectatorMessage.spec` [package]

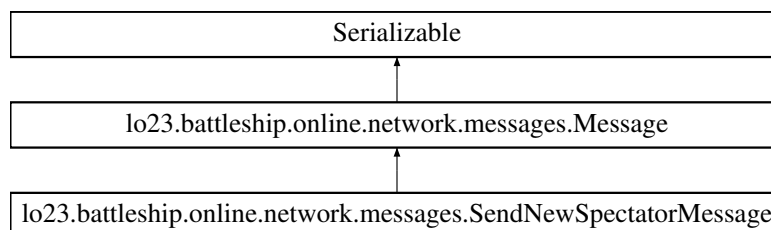
The documentation for this class was generated from the following file:

- `src/main/java/lo23/battleship/online/network/messages/SendInfoGameForSpectatorMessage.java`

7.66 `lo23.battleship.online.network.messages.SendNewSpectatorMessage` Class Reference

This class implements the message sent when a user has been accepted as spectator by the creator of the game.

Inheritance diagram for `lo23.battleship.online.network.messages.SendNewSpectatorMessage`:



Public Member Functions

- `SendNewSpectatorMessage` (`User` s)
Allocates a new.
- `String getType` ()
Returns the type of the message.
- `void process` (`IDataCom` `IData`, `InetAddress` `senderAddress`)
Adds the spectator to the list of spectators of the game using the data package interface `IData`

Private Attributes

- [User spec](#)

Additional Inherited Members

7.66.1 Detailed Description

This class implements the message sent when a user has been accepted as spectator by the creator of the game.

The message is sent to all the current spectators and the other player to notify them of the new spectator's arrival.

See also

[Message](#)

Author

COM Module

7.66.2 Constructor & Destructor Documentation

7.66.2.1 `lo23.battleship.online.network.messages.SendNewSpectatorMessage.SendNewSpectatorMessage (User s)`

Allocates a new.

`SendNewSpectatorMessage`

object

Parameters

s	:
	<code>User</code>
	: new spectator

7.66.3 Member Function Documentation

7.66.3.1 `String lo23.battleship.online.network.messages.SendNewSpectatorMessage.getType ()`

Returns the type of the message.

Implementation of an abstract method.

Returns

type :
`String`

7.66.3.2 `void lo23.battleship.online.network.messages.SendNewSpectatorMessage.process (IDataCom IData, InetAddress senderAddress)`

Adds the spectator to the list of spectators of the game using the data package interface `IData`

Parameters

<code>IData</code>	: <code>IDataCom</code> instance of <code>IDataCom</code> interface.
<code>senderAddress</code>	: <code>InetAddress</code> sender's IP address

7.66.4 Member Data Documentation

7.66.4.1 **User** `lo23.battleship.online.network.messages.SendNewSpectatorMessage.spec` `[private]`

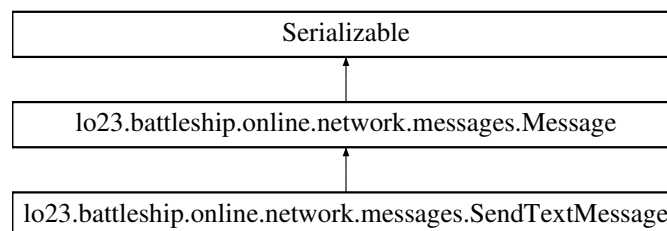
The documentation for this class was generated from the following file:

- `src/main/java/lo23/battleship/online/network/messages/SendNewSpectatorMessage.java`

7.67 `lo23.battleship.online.network.messages.SendTextMessage` Class Reference

This class implements textual messages sent/received by the local user to/from a distant user.

Inheritance diagram for `lo23.battleship.online.network.messages.SendTextMessage`:



Public Member Functions

- `SendTextMessage (ChatMessage m)`
Allocates a new.
- `String getType ()`
Returns the type of the message.
- `void process (IDataCom IData, InetAddress senderAddress)`
Method providing the textual message to data package interface.

Package Attributes

- [ChatMessage message](#)

7.67.1 Detailed Description

This class implements textual messages sent/received by the local user to/from a distant user.

See also

[Message](#)

Author

COM Module

7.67.2 Constructor & Destructor Documentation

7.67.2.1 `lo23.battleship.online.network.messages.SendTextMessage.SendTextMessage (ChatMessage m)`

Allocates a new.

[SendTextMessage](#)

object

Parameters

<i>m</i>	:
	ChatMessage
	: textual message to send.

7.67.3 Member Function Documentation

7.67.3.1 `String lo23.battleship.online.network.messages.SendTextMessage.getType ()`

Returns the type of the message.

Implementation of an abstract method.

Returns

type :

String

7.67.3.2 void `lo23.battleship.online.network.messages.SendTextMessage.process (IDataCom IData, InetAddress senderAddress)`

Method providing the textual message to data package interface.

Parameters

<i>IData</i>	: IDataCom instance of IDataCom interface.
<i>senderAddress</i>	: InetAddress sender's IP address

7.67.4 Member Data Documentation

7.67.4.1 **ChatMessage** `lo23.battleship.online.network.messages.SendTextMessage.message` [package]

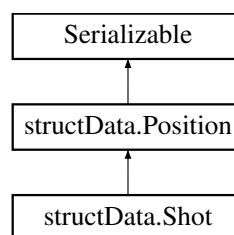
The documentation for this class was generated from the following file:

- `src/main/java/lo23/battleship/online/network/messages/SendTextMessage.java`

7.68 structData.Shot Class Reference

[Shot](#) is a class of a shot of a player.

Inheritance diagram for structData.Shot:



Public Member Functions

- [Shot](#) ([Position](#) p)
Shot : default constructor with a [Position](#).
- [Shot](#) ([Position](#) p, Date nTime)
Shot : constructor with a [Position](#) and a time.
- void [setTime](#) (Date nTime)
Mutator for time.
- Date [getTime](#) ()
Accessor for time.

Private Attributes

- Date [time](#)

Additional Inherited Members

7.68.1 Detailed Description

[Shot](#) is a class of a shot of a player.

7.68.2 Constructor & Destructor Documentation

7.68.2.1 `structData.Shot.Shot (Position p)`

[Shot](#) : default constructor with a [Position](#).

Parameters

<i>p</i>	a Position
----------	----------------------------

7.68.2.2 `structData.Shot.Shot (Position p, Date nTime)`

[Shot](#) : constructor with a [Position](#) and a time.

Parameters

<i>p</i>	a position
<i>nTime</i>	

7.68.3 Member Function Documentation

7.68.3.1 `Date structData.Shot.getTime ()`

Accessor for time.

Returns

time : the shot's time as a Date

7.68.3.2 `void structData.Shot.setTime (Date nTime)`

Mutator for time.

Parameters

<code>nTime</code>	the shot's time
--------------------	-----------------

7.68.4 Member Data Documentation

7.68.4.1 Date `structData.Shot.time` `[private]`

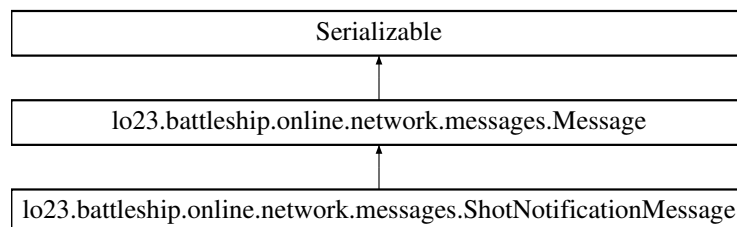
The documentation for this class was generated from the following file:

- `src/main/java/structData/Shot.java`

7.69 `lo23.battleship.online.network.messages.ShotNotificationMessage` Class Reference

This class implements the message sent by the local user (player) when he shots.

Inheritance diagram for `lo23.battleship.online.network.messages.ShotNotificationMessage`:



Public Member Functions

- `ShotNotificationMessage` (`Shot s`)
Allocates a new.
- String `getType` ()
Returns the type of the message.
- void `process` (`IDataCom` `IData`, `InetAddress` `senderAddress`)
Method providing the shot to data package interface to process the shot(missed, touched or sunk boat)

Private Attributes

- `Shot shot`

Additional Inherited Members

7.69.1 Detailed Description

This class implements the message sent by the local user (player) when he shots.

See also

[Message](#)

Author

COM Module

7.69.2 Constructor & Destructor Documentation

7.69.2.1 `io23.battleship.online.network.messages.ShotNotificationMessage.ShotNotificationMessage (Shot s)`

Allocates a new.

`ShotNotificationMessage`

object

Parameters

s	:
	<code>Shot</code>
	the shot of the local user(player)

7.69.3 Member Function Documentation

7.69.3.1 `String io23.battleship.online.network.messages.ShotNotificationMessage.getType ()`

Returns the type of the message.

Implementation of an abstract method.

Returns

type :
`String`

7.69.3.2 `void io23.battleship.online.network.messages.ShotNotificationMessage.process (IDataCom IData, InetAddress senderAddress)`

Method providing the shot to data package interface to process the shot(missed, touched or sunk boat)

Parameters

<i>IData</i>	: <code>IDataCom</code> instance of <code>IDataCom</code> interface.
<i>senderAddress</i>	: <code>InetAddress</code> sender's IP address

7.69.4 Member Data Documentation

7.69.4.1 Shot `lo23.battleship.online.network.messages.ShotNotificationMessage.shot` [private]

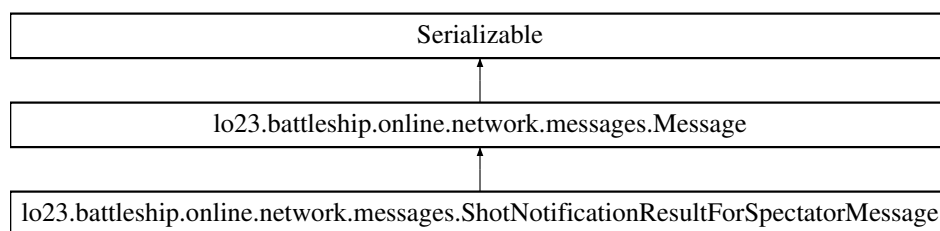
The documentation for this class was generated from the following file:

- `src/main/java/lo23/battleship/online/network/messages/ShotNotificationMessage.java`

7.70 `lo23.battleship.online.network.messages.ShotNotificationResultForSpectator`↔
Message Class Reference

This class implements the message sent by the local user (player) to the spectators with the result of the other player shot (missed, touched or sunk boat).

Inheritance diagram for `lo23.battleship.online.network.messages.ShotNotificationResultForSpectatorMessage`:



Public Member Functions

- `ShotNotificationResultForSpectatorMessage` (`Player` p, `Shot` s, `Boat` b)
Allocates a new.
- String `getType` ()
Returns the type of the message.
- void `process` (`IDataCom` IData, `InetAddress` senderAddress)
Provides the shot result to data package interface to process it (display it eventually).

Private Attributes

- [Shot shot](#)
- [Boat boat](#)
- [Player player](#)

Additional Inherited Members

7.70.1 Detailed Description

This class implements the message sent by the local user (player) to the spectators with the result of the other player shot (missed, touched or sunk boat).

See also

[Message](#)

Author

COM Module

7.70.2 Constructor & Destructor Documentation

7.70.2.1 `Io23.battleship.online.network.messages.ShotNotificationResultForSpectatorMessage.ShotNotificationResultForSpectatorMessage (Player p, Shot s, Boat b)`↔

Allocates a new.

`ShotNotificationResultForSpectatorMessage`

message

Parameters

<i>p</i>	: Player Player who shots
<i>s</i>	: Shot the shot made by the <code>player</code>
<i>b</i>	: Boat the boat sunk by the <code>player</code> (null if no boat has been sunk)

7.70.3 Member Function Documentation

7.70.3.1 `String lo23.battleship.online.network.messages.ShotNotificationResultForSpectatorMessage.getType ()`

Returns the type of the message.

Implementation of an abstract method.

Returns

type :

`String`

7.70.3.2 `void lo23.battleship.online.network.messages.ShotNotificationResultForSpectatorMessage.process (IDataCom IData, InetAddress senderAddress)`

Provides the shot result to data package interface to process it (display it eventually).

Parameters

<i>IData</i>	: <code>IDataCom</code> instance of <code>IDataCom</code> interface.
<i>senderAddress</i>	: <code>InetAddress</code> sender's IP address

7.70.4 Member Data Documentation

7.70.4.1 `Boat lo23.battleship.online.network.messages.ShotNotificationResultForSpectatorMessage.boat` [private]

7.70.4.2 `Player lo23.battleship.online.network.messages.ShotNotificationResultForSpectatorMessage.player` [private]

7.70.4.3 `Shot lo23.battleship.online.network.messages.ShotNotificationResultForSpectatorMessage.shot` [private]

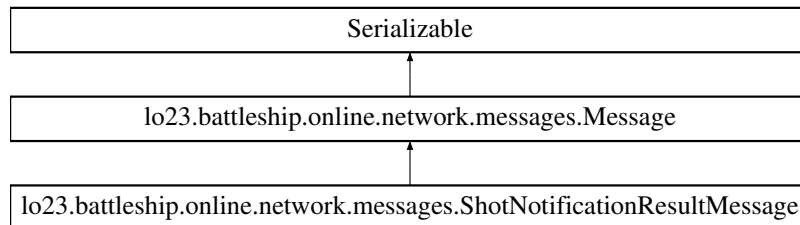
The documentation for this class was generated from the following file:

- `src/main/java/lo23/battleship/online/network/messages/ShotNotificationResultForSpectatorMessage.java`

7.71 `lo23.battleship.online.network.messages.ShotNotificationResultMessage` Class Reference

This class implements the message sent by the local user (player) to the other player with the result of the other player's shot (missed, touched or sunk boat).

Inheritance diagram for `lo23.battleship.online.network.messages.ShotNotificationResultMessage`:



Public Member Functions

- [ShotNotificationResultMessage](#) ([Shot](#) s, [Boat](#) b)
Allocates a new.
- [String](#) [getType](#) ()
Returns the type of the message.
- [void](#) [process](#) ([IDataCom](#) IData, [InetAddress](#) senderAddress)
Provides the shot result and the boat to data package interface to process it (display it eventually).

Private Attributes

- [Shot](#) shot
- [Boat](#) boat

Additional Inherited Members

7.71.1 Detailed Description

This class implements the message sent by the local user (player) to the other player with the result of the other player's shot (missed, touched or sunk boat).

See also

[Message](#)

Author

COM Module

7.71.2 Constructor & Destructor Documentation

7.71.2.1 `lo23.battleship.online.network.messages.ShotNotificationResultMessage.ShotNotificationResultMessage (Shot s, Boat b)`

Allocates a new.

[ShotNotificationResultMessage](#)

message

Parameters

s	:	<div>Shot</div> <div>the shot made by the other player made</div>
b	:	<div>Boat</div> <div>the boat sunk by the other player (null if no boat has been sunk)</div>

7.71.3 Member Function Documentation

7.71.3.1 String io23.battleship.online.network.messages.ShotNotificationResultMessage.getType ()

Returns the type of the message.

Implementation of an abstract method.

Returns

type :

String

7.71.3.2 void io23.battleship.online.network.messages.ShotNotificationResultMessage.process (IDataCom IData, InetAddress senderAddress)

Provides the shot result and the boat to data package interface to process it (display it eventually).

Parameters

IData	:	<div>IDataCom</div> <div>instance of IDataCom interface.</div>
senderAddress	:	<div>InetAddress</div> <div>sender's IP address</div>

7.71.4 Member Data Documentation

7.71.4.1 Boat io23.battleship.online.network.messages.ShotNotificationResultMessage.boat [private]

7.71.4.2 Shot io23.battleship.online.network.messages.ShotNotificationResultMessage.shot [private]

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

- [src/main/java/lo23/battleship/online/network/messages/ShotNotificationResultMessage.java](#)

7.72 guiMain.controller.SignupController Class Reference

This class implements the controller of the SignUp page.

Public Member Functions

- void [setMainController](#) ([GuiMainController](#) c)
Set [GuiMainController](#) mainController.
- void [init](#) ()
Called at initialization.

Private Member Functions

- void [chooseAvatar](#) (ActionEvent event)
Open a File Chooser pop up.
- void [createAccount](#) ()
Create a new account.
- void [addConnectionPoint](#) ()
Open the Ip configuration page.
- void [backToConnectionWindow](#) ()
Return to the starting page (Connection page)
- void [openFile](#) (File file)
Open the image file and set the avatar to this image.

Private Attributes

- [GuiMainController](#) [mainController](#)
- [ImageView](#) [avatarImage](#)
- [Button](#) [chooseAvatarButton](#)
- [Button](#) [addConnectionButton](#)
- [Button](#) [createButton](#)
- [Button](#) [connectionPageButton](#)
- [TextField](#) [idTextField](#)
- [PasswordField](#) [passwordTextField](#)
- [TextField](#) [usernameTextField](#)
- [TextField](#) [firstNameTextField](#)
- [TextField](#) [lastNameTextField](#)
- [DatePicker](#) [birthDatePicker](#)
- [Label](#) [messageLabel](#)
- [String](#) [avatarPath](#)

7.72.1 Detailed Description

This class implements the controller of the SignUp page.

Author

IHM-Main module

7.72.2 Member Function Documentation

7.72.2.1 void guiMain.controller.SignupController.addConnectionPoint () [private]

Open the Ip configuration page.

7.72.2.2 void guiMain.controller.SignupController.backToConnectionWindow () [private]

Return to the starting page (Connection page)

7.72.2.3 void guiMain.controller.SignupController.chooseAvatar (ActionEvent *event*) [private]

Open a File Chooser pop up.

Parameters

<i>event</i>	: #validateButton
--------------	-------------------

7.72.2.4 void guiMain.controller.SignupController.createAccount () [private]

Create a new account.

7.72.2.5 void guiMain.controller.SignupController.init ()

Called at initialization.

7.72.2.6 void guiMain.controller.SignupController.openFile (File *file*) [private]

Open the image file and set the avatar to this image.

7.72.2.7 void guiMain.controller.SignupController.setMainController (GuiMainController *c*)

Set [GuiMainController](#) mainController.

Parameters

<i>c</i>	: GuiMainController
----------	-------------------------------------

7.72.3 Member Data Documentation

7.72.3.1 Button guiMain.controller.SignupController.addConnectionButton [private]

- 7.72.3.2 `ImageView guiMain.controller.SignupController.avatarImage` `[private]`
- 7.72.3.3 `String guiMain.controller.SignupController.avatarPath` `[private]`
- 7.72.3.4 `DatePicker guiMain.controller.SignupController.birthDatePicker` `[private]`
- 7.72.3.5 `Button guiMain.controller.SignupController.chooseAvatarButton` `[private]`
- 7.72.3.6 `Button guiMain.controller.SignupController.connectionPageButton` `[private]`
- 7.72.3.7 `Button guiMain.controller.SignupController.createButton` `[private]`
- 7.72.3.8 `TextField guiMain.controller.SignupController.firstNameTextField` `[private]`
- 7.72.3.9 `TextField guiMain.controller.SignupController.idTextField` `[private]`
- 7.72.3.10 `TextField guiMain.controller.SignupController.lastNameTextField` `[private]`
- 7.72.3.11 `GuiMainController guiMain.controller.SignupController.mainController` `[private]`
- 7.72.3.12 `Label guiMain.controller.SignupController.messageLabel` `[private]`
- 7.72.3.13 `PasswordField guiMain.controller.SignupController.passwordTextField` `[private]`
- 7.72.3.14 `TextField guiMain.controller.SignupController.usernameTextField` `[private]`

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

- [src/main/java/guiMain/controller/SignupController.java](#)

7.73 structData.StatusGame Enum Reference

[StatusGame](#) is an enumerate of all the status of games.

Public Attributes

- [WAITINGPLAYER](#)
- [WAITINGBOT](#)
- [BOATPHASE](#)
- [PLAYER1READY](#)
- [PLAYER2READY](#)
- [PLAYING](#)
- [FINISHED](#)

7.73.1 Detailed Description

[StatusGame](#) is an emumerate of all the status of games.

7.73.2 Member Data Documentation

7.73.2.1 `structData.StatusGame.BOATPHASE`

7.73.2.2 `structData.StatusGame.FINISHED`

7.73.2.3 `structData.StatusGame.PLAYER1READY`

7.73.2.4 `structData.StatusGame.PLAYER2READY`

7.73.2.5 `structData.StatusGame.PLAYING`

7.73.2.6 `structData.StatusGame.WAITINGBOT`

7.73.2.7 `structData.StatusGame.WAITINGPLAYER`

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

- `src/main/java/structData/StatusGame.java`

7.74 guiTable.CaseDrawing.Type Enum Reference

Public Attributes

- [TOUCHED](#)
- [MISSED](#)
- [SHOT](#)
- [BOAT](#)
- [SUNK_BOAT](#)

7.74.1 Member Data Documentation

7.74.1.1 `guiTable.CaseDrawing.Type.BOAT`

7.74.1.2 `guiTable.CaseDrawing.Type.MISSED`

7.74.1.3 `guiTable.CaseDrawing.Type.SHOT`

7.74.1.4 `guiTable.CaseDrawing.Type.SUNK_BOAT`

7.74.1.5 `guiTable.CaseDrawing.Type.TOUCHED`

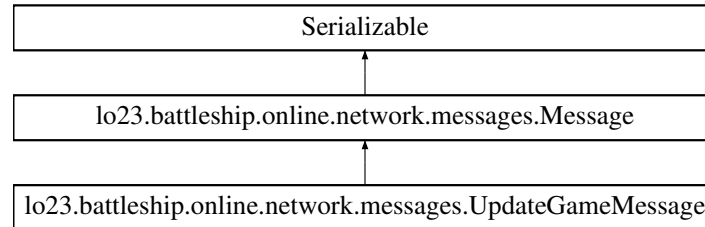
The documentation for this enum was generated from the following file:

- `src/main/java/guiTable/CaseDrawing.java`

7.75 Io23.battleship.online.network.messages.UpdateGameMessage Class Reference

This class implements the message sent by the local user to other users when a game (namely the status of the game) has changed.

Inheritance diagram for Io23.battleship.online.network.messages.UpdateGameMessage:



Public Member Functions

- [UpdateGameMessage](#) ([Game](#) game)
Allocates a new.
- String [getType](#) ()
Message type getter.
- void [process](#) ([IDataCom](#) IData, [InetAddress](#) senderAddress)
Provides the updated game to Data.

Private Attributes

- [Game](#) gameUpdate

Additional Inherited Members

7.75.1 Detailed Description

This class implements the message sent by the local user to other users when a game (namely the status of the game) has changed.

See also

[Message](#)

Author

COM Module

7.75.2 Constructor & Destructor Documentation

7.75.2.1 Io23.battleship.online.network.messages.UpdateGameMessage.UpdateGameMessage ([Game](#) game)

Allocates a new.

[UpdateGameMessage](#)

object

Parameters

<i>game</i>	:	
		Message
		game which has been updated.

7.75.3 Member Function Documentation

7.75.3.1 `String lo23.battleship.online.network.messages.UpdateGameMessage.getType ()`

[Message](#) type getter.

Implementation of an abstract method.

Returns

type, this is the message type.

7.75.3.2 `void lo23.battleship.online.network.messages.UpdateGameMessage.process (IDataCom IData, InetAddress senderAddress)`

Provides the updated game to Data.

Parameters

<i>IData</i>	:	
		IDataCom
		instance of IDataCom interface.
<i>senderAddress</i>	:	
		InetAddress
		sender's IP address

7.75.4 Member Data Documentation

7.75.4.1 `Game lo23.battleship.online.network.messages.UpdateGameMessage.gameUpdate` `[private]`

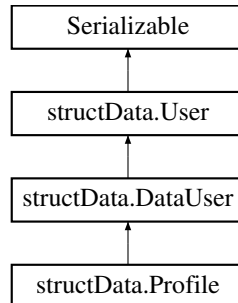
The documentation for this class was generated from the following file:

- `src/main/java/lo23/battleship/online/network/messages/UpdateGameMessage.java`

7.76 structData.User Class Reference

Class [User](#).

Inheritance diagram for structData.User:



Public Member Functions

- [User](#) (String loginData, String usernameData)
Constructor with parameters.
- [User](#) ([User](#) u)
Constructor for the class [DataUser](#) who is "son" of the class [User](#).
- UUID [getIdUser](#) ()
Accessors for idUser.
- String [getLogin](#) ()
Accessors for login.
- String [getUsername](#) ()
Accessors for Username.
- HashSet [getIPs](#) ()
Accessors for iPs.
- void [setIdUser](#) (UUID idUserData)
Mutator for idUser.
- void [setLogin](#) (String loginData)
Mutator for login.
- void [setUsername](#) (String usernameData)
Mutator for userName.
- void [setIPs](#) (HashSet iPsData)
Mutator for iPs.
- [User](#) [cloneUser](#) ([User](#) userClone)
Clone function to copy a [User](#).
- void [setPort](#) (int p)
Mutator for the port.
- int [getPort](#) ()
Accessor for the [User](#)'s port.

Static Public Attributes

- static final int [DEFAULT_PORT](#) = 2345

Protected Attributes

- UUID [idUser](#)
- String [login](#)
- String [username](#)
- HashSet< InetAddress > [iPs](#)
- int [port](#) = [DEFAULT_PORT](#)

Static Package Attributes

- static final long [serialVersionUID](#) = 7L

7.76.1 Detailed Description

Class [User](#).

7.76.2 Constructor & Destructor Documentation

7.76.2.1 structData.User.User (String *loginData*, String *usernameData*)

Constructor with parameters.

Parameters

<i>loginData</i>	: the login of the new user
<i>usernameData</i>	: the username of the new user

7.76.2.2 structData.User.User (User *u*)

Constructor for the class [DataUser](#) who is "son" of the class [User](#).

Parameters

<i>u</i>	: a new object of the class User
----------	--

7.76.3 Member Function Documentation

7.76.3.1 User structData.User.cloneUser (User *userClone*)

Clone function to copy a [User](#).

Parameters

<i>userClone</i>	: the user to be cloned
------------------	-------------------------

Returns

the user who has called this method to clone all the information of the userClone

7.76.3.2 UUID structData.User.getIdUser ()

Accessors for idUser.

Returns

the id of the user

7.76.3.3 HashSet structData.User.getIPs ()

Accessors for iPs.

Returns

iPs of the user

7.76.3.4 String structData.User.getLogin ()

Accessors for login.

Returns

the login of the user

7.76.3.5 int structData.User.getPort ()

Accessor for the [User](#)'s port.

Returns

the [User](#)'s port

7.76.3.6 String structData.User.getUsername ()

Accessors for Username.

Returns

the username of the user

7.76.3.7 void structData.User.setIdUser (UUID *idUserData*)

Mutator for idUser.

Parameters

<i>idUserData</i>	: the new value of the id of the user
-------------------	---------------------------------------

7.76.3.8 void structData.User.setIPs (HashSet *iPsData*)

Mutator for iPs.

Parameters

<i>iPsData</i>	: the new value of the iPs of the user
----------------	--

7.76.3.9 void structData.User.setLogin (String *loginData*)

Mutator for login.

Parameters

<i>loginData</i>	: the new value of the login of the user
------------------	--

7.76.3.10 void structData.User.setPort (int *p*)

Mutator for the port.

Parameters

<i>p</i>	: port
----------	--------

7.76.3.11 void structData.User.setUsername (String *usernameData*)

Mutator for userName.

Parameters

<i>usernameData</i>	: the new value of the username of the user
---------------------	---

7.76.4 Member Data Documentation

7.76.4.1 final int structData.User.DEFAULT_PORT = 2345 [static]

7.76.4.2 UUID structData.User.idUser [protected]

7.76.4.3 `HashSet<InetAddress> structData.User.iPs` [protected]

7.76.4.4 `String structData.User.login` [protected]

7.76.4.5 `int structData.User.port = DEFAULT_PORT` [protected]

7.76.4.6 `final long structData.User serialVersionUID = 7L` [static], [package]

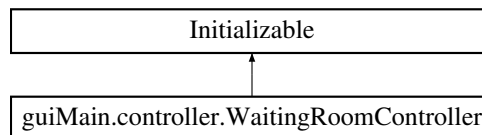
7.76.4.7 `String structData.User.username` [protected]

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

- `src/main/java/structData/User.java`

7.77 guiMain.controller.WaitingRoomController Class Reference

Inheritance diagram for guiMain.controller.WaitingRoomController:



Public Member Functions

- void `initialize` (URL location, ResourceBundle resources)
- void `initData` (`Game` game)
Initialize window elements with game data.
- void `closeWindow` ()
Close waiting room window.
- void `setStage` (Stage s)
Set stage.

Private Attributes

- Label `gameName`
- Label `gameType`
- Label `spectatorAuthorisation`
- Label `chatAuthorisation`
- Label `reflectionTime`
- Stage `stage`

7.77.1 Member Function Documentation

7.77.1.1 `void guiMain.controller.WaitingRoomController.closeWindow ()`

Close waiting room window.

7.77.1.2 `void guiMain.controller.WaitingRoomController.initData (Game game)`

Initialize window elements with game data.

Parameters

<i>game</i>	
-------------	--

7.77.1.3 void guiMain.controller.WaitingRoomController.initialize (URL *location*, ResourceBundle *resources*)

7.77.1.4 void guiMain.controller.WaitingRoomController.setStage (Stage *s*)

Set stage.

Parameters

<i>s</i>	
----------	--

7.77.2 Member Data Documentation

7.77.2.1 Label guiMain.controller.WaitingRoomController.chatAutorisation [private]

7.77.2.2 Label guiMain.controller.WaitingRoomController.gameName [private]

7.77.2.3 Label guiMain.controller.WaitingRoomController.gameType [private]

7.77.2.4 Label guiMain.controller.WaitingRoomController.reflectionTime [private]

7.77.2.5 Label guiMain.controller.WaitingRoomController.spectatorAutorisation [private]

7.77.2.6 Stage guiMain.controller.WaitingRoomController.stage [private]

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

- src/main/java/guiMain/controller/[WaitingRoomController.java](#)

Chapter 8

File Documentation

8.1 README.md File Reference

8.2 src/main/java/data/CDataCom.java File Reference

Classes

- class [data.CDataCom](#)
CDataCom : interface of Data for Com.

Packages

- package [data](#)

8.3 src/main/java/data/CDataMain.java File Reference

Classes

- class [data.CDataMain](#)
CDataCom : interface of Data for IHM Main.

Packages

- package [data](#)

8.4 src/main/java/data/CDataTable.java File Reference

Classes

- class [data.CDataTable](#)
Data's interface for Table.

Packages

- package [data](#)

8.5 [src/main/java/data/DataController.java](#) File Reference

Classes

- class [data.DataController](#)
Data's controller.

Packages

- package [data](#)

8.6 [src/main/java/guiMain/controller/ChangeProfileController.java](#) File Reference

Classes

- class [guiMain.controller.ChangeProfileController](#)

Packages

- package [guiMain.controller](#)

8.7 [src/main/java/guiMain/controller/CreateGameController.java](#) File Reference

Classes

- class [guiMain.controller.CreateGameController](#)

Packages

- package [guiMain.controller](#)

8.8 [src/main/java/guiMain/controller/IpConfigController.java](#) File Reference

Classes

- class [guiMain.controller.IpConfigController](#)
This class implements the controller of the Ip Configuration page.

Packages

- package [guiMain.controller](#)

8.9 src/main/java/guiMain/controller/LoginController.java File Reference

Classes

- class [guiMain.controller.LoginController](#)

Packages

- package [guiMain.controller](#)

8.10 src/main/java/guiMain/controller/menuController.java File Reference

Classes

- class [guiMain.controller.menuController](#)

This class call the lobby view and enable interaction between button and method.

Packages

- package [guiMain.controller](#)

8.11 src/main/java/guiMain/controller/ProfilController.java File Reference

Classes

- class [guiMain.controller.ProfilController](#)

*[GameCell](#), implement interface *Initializable*, display the user profil.*

Packages

- package [guiMain.controller](#)

8.12 src/main/java/guiMain/controller/SignupController.java File Reference

Classes

- class [guiMain.controller.SignupController](#)

*This class implements the controller of the *SignUp* page.*

Packages

- package [guiMain.controller](#)

8.13 src/main/java/guiMain/controller/WaitingRoomController.java File Reference

Classes

- class [guiMain.controller.WaitingRoomController](#)

Packages

- package [guiMain.controller](#)

8.14 src/main/java/guiMain/GameCell.java File Reference

Classes

- class [guiMain.GameCell](#)
[GameCell](#), descendant class [ListCell<Game>](#), display the game name and offers the possibility to join or watch a game.

Packages

- package [guiMain](#)

8.15 src/main/java/guiMain/GuiMainController.java File Reference

Classes

- class [guiMain.GuiMainController](#)
This class implements the network controller.

Packages

- package [guiMain](#)

8.16 src/main/java/guiMain/GuiMainInterface.java File Reference

Classes

- interface [guiMain.GuiMainInterface](#)
GUI Main Interface.

Packages

- package [guiMain](#)

8.17 src/main/java/guiMain/IpCell.java File Reference

Classes

- class [guiMain.IpCell](#)
[IpCell](#), Display a list of Ips on a list that can be removed with a button.

Packages

- package [guiMain](#)

8.18 src/main/java/guiMain/PlayerCell.java File Reference

Classes

- class [guiMain.PlayerCell](#)
[PlayerCell](#), descendant class [ListCell<User>](#), display the user name and offers the possibility watch his profil.

Packages

- package [guiMain](#)

8.19 src/main/java/guiTable/BoatDrawing.java File Reference

Classes

- class [guiTable.BoatDrawing](#)
Boat drawing : class to draw a boat.

Packages

- package [guiTable](#)

8.20 src/main/java/guiTable/CaseDrawing.java File Reference

Classes

- class [guiTable.CaseDrawing](#)
[CaseDrawing](#) : class to draw a case.
- enum [guiTable.CaseDrawing.Type](#)

Packages

- package [guiTable](#)

8.21 src/main/java/guiTable/controllers/BaseController.java File Reference

Classes

- class [guiTable.controllers.BaseController](#)
every class controller will herit from [BaseController](#)

Packages

- package [guiTable.controllers](#)

8.22 src/main/java/guiTable/controllers/BelgianPlacementPhaseController.java File Reference

Classes

- class [guiTable.controllers.BelgianPlacementPhaseController](#)
implementation of placement controller for belgian phase

Packages

- package [guiTable.controllers](#)

8.23 src/main/java/guiTable/controllers/ChatController.java File Reference

Classes

- class [guiTable.controllers.ChatController](#)

Packages

- package [guiTable.controllers](#)

8.24 src/main/java/guiTable/controllers/ClassicPlacementPhaseController.java File Reference

Classes

- class [guiTable.controllers.ClassicPlacementPhaseController](#)
implementation of placement controller for classic phase

Packages

- package [guiTable.controllers](#)

8.25 src/main/java/guiTable/controllers/gameInterface.java File Reference

Classes

- class [guiTable.controllers.gameInterface](#)
gameInterface

Packages

- package [guiTable.controllers](#)

8.26 src/main/java/guiTable/controllers/GamePhaseController.java File Reference

Classes

- class [guiTable.controllers.GamePhaseController](#)
GamePhaseController.

Packages

- package [guiTable.controllers](#)

8.27 src/main/java/guiTable/controllers/GuiTableController.java File Reference

Classes

- class [guiTable.controllers.GuiTableController](#)
GuiTableController.

Packages

- package [guiTable.controllers](#)

8.28 src/main/java/guiTable/controllers/ObservationPhase.java File Reference

Classes

- class [guiTable.controllers.ObservationPhase](#)

Packages

- package [guiTable.controllers](#)

8.29 src/main/java/guiTable/controllers/ObserverPhaseController.java File Reference

Classes

- class [guiTable.controllers.ObserverPhaseController](#)
ObserverPhaseController.

Packages

- package [guiTable.controllers](#)

8.30 src/main/java/guiTable/controllers/PlacementPhaseController.java File Reference

Classes

- class [guiTable.controllers.PlacementPhaseController](#)
PlacementPhaseController.

Packages

- package [guiTable.controllers](#)

8.31 src/main/java/guiTable/controllers/ProfileController.java File Reference

Classes

- class [guiTable.controllers.ProfileController](#)

Packages

- package [guiTable.controllers](#)

8.32 src/main/java/guiTable/GuiTableInterface.java File Reference

Classes

- interface [guiTable.GuiTableInterface](#)
Interface for the Ihm Table team.

Packages

- package [guiTable](#)

8.33 src/main/java/interfacesData/IDataCom.java File Reference

Classes

- interface [interfacesData.IDataCom](#)
Data's interface for Com.

Packages

- package [interfacesData](#)

8.34 src/main/java/interfacesData/IDataMain.java File Reference

Classes

- interface [interfacesData.IDataMain](#)
Data's interface for Main.

Packages

- package [interfacesData](#)

8.35 src/main/java/interfacesData/IDataTable.java File Reference

Classes

- interface [interfacesData.IDataTable](#)
Data's interface for IHM-Table.

Packages

- package [interfacesData](#)

8.36 src/main/java/lo23/battleship/MainApp.java File Reference

Classes

- class [lo23.battleship.MainApp](#)

Packages

- package [lo23.battleship](#)

8.37 src/main/java/lo23/battleship/online/network/COMInterface.java File Reference

Classes

- interface [lo23.battleship.online.network.COMInterface](#)
The [COMInterface](#) compiles the services offered by the network package.

Packages

- package [lo23.battleship.online.network](#)
Created by xzirva on 17/10/17.

8.38 src/main/java/lo23/battleship/online/network/messages/ConnectionEstablishedMessage.java File Reference

Classes

- class [lo23.battleship.online.network.messages.ConnectionEstablishedMessage](#)
This class implements the message which is sent when when a user disconnects from the application.

Packages

- package [lo23.battleship.online.network.messages](#)
Created by xzirva on 17/10/17.

8.39 src/main/java/lo23/battleship/online/network/messages/ConnectionRequestMessage.java File Reference

Classes

- class [lo23.battleship.online.network.messages.ConnectionRequestMessage](#)
Network message class, extends [Message](#) class.

Packages

- package [lo23.battleship.online.network.messages](#)
Created by xzirva on 17/10/17.

8.40 src/main/java/lo23/battleship/online/network/messages/CreatedGameNotificationMessage.java File Reference

Classes

- class [lo23.battleship.online.network.messages.CreatedGameNotificationMessage](#)
Network message class, extends [Message](#) class.

Packages

- package [lo23.battleship.online.network.messages](#)
Created by xzirva on 17/10/17.

8.41 src/main/java/lo23/battleship/online/network/messages/DisconnectionMessage.java File Reference

Classes

- class [lo23.battleship.online.network.messages.DisconnectionMessage](#)
This class implements the message which is sent when when a user disconnects from the application.

Packages

- package [lo23.battleship.online.network.messages](#)
Created by xzirva on 17/10/17.

8.42 src/main/java/lo23/battleship/online/network/messages/GameQuitMessage.java File Reference

Classes

- class [lo23.battleship.online.network.messages.GameQuitMessage](#)
This class implements the message sent when a game is removed from the network (and the list of games) by his creator(user) or when the game is over.

Packages

- package [lo23.battleship.online.network.messages](#)
Created by xzirva on 17/10/17.

8.43 `src/main/java/lo23/battleship/online/network/messages/GameQuitSpectatorMessage.java` File Reference

Classes

- class [lo23.battleship.online.network.messages.GameQuitSpectatorMessage](#)
This class implements the message which is sent when a spectator quit a game he is watching.

Packages

- package [lo23.battleship.online.network.messages](#)
Created by xzirva on 17/10/17.

8.44 `src/main/java/lo23/battleship/online/network/messages/GameWonMessage.java` File Reference

Classes

- class [lo23.battleship.online.network.messages.GameWonMessage](#)
This class implements the message which is sent when a game is over with a victory of one of the players.

Packages

- package [lo23.battleship.online.network.messages](#)
Created by xzirva on 17/10/17.

8.45 `src/main/java/lo23/battleship/online/network/messages/GameWonMessageToSpectator.java` File Reference

Classes

- class [lo23.battleship.online.network.messages.GameWonMessageToSpectator](#)
This class implements the message which is sent when a game is over with a victory of one of the players.

Packages

- package [lo23.battleship.online.network.messages](#)
Created by xzirva on 17/10/17.

8.46 src/main/java/lo23/battleship/online/network/messages/GetInfoGameForSpectatorMessage.java File Reference ↩

Classes

- class [lo23.battleship.online.network.messages.GetInfoGameForSpectatorMessage](#)
This class implements the message which is sent when when a user wants to watch a game.

Packages

- package [lo23.battleship.online.network.messages](#)
Created by xzirva on 17/10/17.

8.47 src/main/java/lo23/battleship/online/network/messages/GetProfileRequestMessage.java File Reference

Classes

- class [lo23.battleship.online.network.messages.GetProfileRequestMessage](#)
This class implements the message which is sent when when a user wants to see the profile(including statistics) of an other user.

Packages

- package [lo23.battleship.online.network.messages](#)
Created by xzirva on 17/10/17.

8.48 src/main/java/lo23/battleship/online/network/messages/JoinGameRequestMessage.java File Reference

Classes

- class [lo23.battleship.online.network.messages.JoinGameRequestMessage](#)
This class implements the message which is sent when when a user wants to join a game created by another user.

Packages

- package [lo23.battleship.online.network.messages](#)
Created by xzirva on 17/10/17.

8.49 [src/main/java/lo23/battleship/online/network/messages/JoinGameResponseMessage.java](#) File Reference

Classes

- class [lo23.battleship.online.network.messages.JoinGameResponseMessage](#)
This class implements the message which is sent when when a creator of a game adds a second player to the game.

Packages

- package [lo23.battleship.online.network.messages](#)
Created by xzirva on 17/10/17.

8.50 [src/main/java/lo23/battleship/online/network/messages/Message.java](#) File Reference

Classes

- class [lo23.battleship.online.network.messages.Message](#)
This abstract class implements the core structure of a message sent on the network.

Packages

- package [lo23.battleship.online.network.messages](#)
Created by xzirva on 17/10/17.

8.51 [src/main/java/lo23/battleship/online/network/messages/NotifyReadyMessage.java](#) File Reference

Classes

- class [lo23.battleship.online.network.messages.NotifyReadyMessage](#)
This class implements the message sent by the local user (player) to the other player when he is ready to play.

Packages

- package [lo23.battleship.online.network.messages](#)
Created by xzirva on 17/10/17.

8.52 src/main/java/lo23/battleship/online/network/messages/package-info.java File Reference

Packages

- package [lo23.battleship.online.network.messages](#)

Created by xzirva on 17/10/17.

8.53 src/main/java/lo23/battleship/online/network/package-info.java File Reference

Packages

- package [lo23.battleship.online.network](#)

Created by xzirva on 17/10/17.

8.54 src/main/java/lo23/battleship/online/network/messages/ProfileRequestAnswerMessage.java File Reference ↩

Classes

- class [lo23.battleship.online.network.messages.ProfileRequestAnswerMessage](#)

Network message class, extends [Message](#) class.

Packages

- package [lo23.battleship.online.network.messages](#)

Created by xzirva on 17/10/17.

8.55 src/main/java/lo23/battleship/online/network/messages/SendInfoGameForSpectatorMessage.java File Reference ↩

Classes

- class [lo23.battleship.online.network.messages.SendInfoGameForSpectatorMessage](#)

This class implements the message sent when the local user has been accepted as spectator.

Packages

- package [lo23.battleship.online.network.messages](#)

Created by xzirva on 17/10/17.

8.56 `src/main/java/lo23/battleship/online/network/messages/SendNewSpectatorMessage.java` File Reference

Classes

- class [lo23.battleship.online.network.messages.SendNewSpectatorMessage](#)
This class implements the message sent when a user has been accepted as spectator by the creator of the game.

Packages

- package [lo23.battleship.online.network.messages](#)
Created by xzirva on 17/10/17.

8.57 `src/main/java/lo23/battleship/online/network/messages/SendTextMessage.java` File Reference

Classes

- class [lo23.battleship.online.network.messages.SendTextMessage](#)
This class implements textual messages sent/received by the local user to/from a distant user.

Packages

- package [lo23.battleship.online.network.messages](#)
Created by xzirva on 17/10/17.

8.58 `src/main/java/lo23/battleship/online/network/messages/ShotNotificationMessage.java` File Reference

Classes

- class [lo23.battleship.online.network.messages.ShotNotificationMessage](#)
This class implements the message sent by the local user (player) when he shots.

Packages

- package [lo23.battleship.online.network.messages](#)
Created by xzirva on 17/10/17.

8.59

src/main/java/lo23/battleship/online/network/messages/ShotNotificationResultForSpectatorMessage.java

File Reference

273

8.59 src/main/java/lo23/battleship/online/network/messages/ShotNotificationResultForSpectatorMessage.java File Reference ↩

Classes

- class [lo23.battleship.online.network.messages.ShotNotificationResultForSpectatorMessage](#)

This class implements the message sent by the local user (player) to the spectators with the result of the other player shot (missed, touched or sunk boat).

Packages

- package [lo23.battleship.online.network.messages](#)

Created by xzirva on 17/10/17.

8.60 src/main/java/lo23/battleship/online/network/messages/ShotNotificationResultMessage.java File Reference ↩

Classes

- class [lo23.battleship.online.network.messages.ShotNotificationResultMessage](#)

This class implements the message sent by the local user (player) to the other player with the result of the other player's shot (missed, touched or sunk boat).

Packages

- package [lo23.battleship.online.network.messages](#)

Created by xzirva on 17/10/17.

8.61 src/main/java/lo23/battleship/online/network/messages/UpdateGameMessage.java File Reference

Classes

- class [lo23.battleship.online.network.messages.UpdateGameMessage](#)

This class implements the message sent by the local user to other users when a game (namely the status of the game) has changed.

Packages

- package [lo23.battleship.online.network.messages](#)

Created by xzirva on 17/10/17.

8.62 src/main/java/lo23/battleship/online/network/NetworkController.java File Reference

Classes

- class [lo23.battleship.online.network.NetworkController](#)

This class implements the network controller.

Packages

- package [lo23.battleship.online.network](#)

Created by xzirva on 17/10/17.

8.63 src/main/java/lo23/battleship/online/network/NetworkListener.java File Reference

Classes

- class [lo23.battleship.online.network.NetworkListener](#)

This class implements network listeners.

Packages

- package [lo23.battleship.online.network](#)

Created by xzirva on 17/10/17.

8.64 src/main/java/lo23/battleship/online/network/NetworkModuleInterface.java File Reference

Classes

- class [lo23.battleship.online.network.NetworkModuleInterface](#)

This class implements the [COMInterface](#) interface and the different methods (services) it offers.

Packages

- package [lo23.battleship.online.network](#)

Created by xzirva on 17/10/17.

8.65 src/main/java/lo23/battleship/online/network/NetworkSender.java File Reference

Classes

- class [lo23.battleship.online.network.NetworkSender](#)

This class implements network senders.

Packages

- package [lo23.battleship.online.network](#)

Created by xzirva on 17/10/17.

8.66 src/main/java/lo23/battleship/online/network/NetworkServer.java File Reference

Classes

- class [lo23.battleship.online.network.NetworkServer](#)

This class implements the network server.

Packages

- package [lo23.battleship.online.network](#)

Created by xzirva on 17/10/17.

8.67 src/main/java/structData/Boat.java File Reference

Classes

- class [structData.Boat](#)

[Boat](#) is the class which represents a ship of a player.

Packages

- package [structData](#)

8.68 src/main/java/structData/BoatType.java File Reference

Classes

- enum [structData.BoatType](#)

[BoatType](#) is an enumerate of all the types of boats and numbur of cases associated.

Packages

- package [structData](#)

8.69 src/main/java/structData/ChatMessage.java File Reference

Classes

- class [structData.ChatMessage](#)
ChatMessage is a class for chat messages.

Packages

- package [structData](#)

8.70 src/main/java/structData/ContactGroup.java File Reference

Classes

- class [structData.ContactGroup](#)
ContactGroup is a class a contact groups.

Packages

- package [structData](#)

8.71 src/main/java/structData/DataUser.java File Reference

Classes

- class [structData.DataUser](#)
DataUser class : inherits from User.

Packages

- package [structData](#)

8.72 src/main/java/structData/Game.java File Reference

Classes

- class [structData.Game](#)
Game is a class for the user's game.

Packages

- package [structData](#)

8.73 src/main/java/structData/MessageType.java File Reference

Classes

- enum [structData.MessageType](#)
Enumerate for the types of messages (Class Message)

Packages

- package [structData](#)

8.74 src/main/java/structData/Player.java File Reference

Classes

- class [structData.Player](#)
Player is a class for players in a game.

Packages

- package [structData](#)

8.75 src/main/java/structData/Position.java File Reference

Classes

- class [structData.Position](#)
Position is a class of a position in a table.

Packages

- package [structData](#)

8.76 src/main/java/structData/Profile.java File Reference

Classes

- class [structData.Profile](#)
Profile is a class for the user's profile.

Packages

- package [structData](#)

8.77 src/main/java/structData/Shot.java File Reference

Classes

- class [structData.Shot](#)
Shot is a class of a shot of a player.

Packages

- package [structData](#)

8.78 src/main/java/structData/StatusGame.java File Reference

Classes

- enum [structData.StatusGame](#)
StatusGame is an emumerate of all the status of games.

Packages

- package [structData](#)

8.79 src/main/java/structData/User.java File Reference

Classes

- class [structData.User](#)
Class User.

Packages

- package [structData](#)

Index

activateBoat
 guiTable::controllers::PlacementPhaseController, 202

active
 guiTable::BoatDrawing, 29

activeBoat
 guiTable::controllers::PlacementPhaseController, 207

activeColor
 guiTable::BoatDrawing, 29

addBoat
 structData::Player, 209

addButton
 guiMain::controller::lpConfigController, 155

addChatMessage
 guiTable::GuiTableInterface, 134
 guiTable::controllers::GuiTableController, 130

addConnectionButton
 guiMain::controller::SignupController, 245

addConnectionPoint
 guiMain::controller::SignupController, 245

addGame
 guiMain::GuiMainController, 121
 guiMain::GuiMainInterface, 126
 guiMain::controller::menuController, 164

addGameToList
 data::DataController, 77

addIp
 guiMain::controller::lpConfigController, 155

addMessage
 structData::Game, 93

addNewGameList
 data::CDataCom, 34
 interfacesData::IDataCom, 138

addOpponentShot
 guiTable::controllers::GamePhaseController, 105

addPosition
 structData::Boat, 24

addShot
 guiTable::controllers::GamePhaseController, 105
 structData::Player, 210

addSpectators
 structData::Game, 94

addToNetwork
 lo23::battleship::online::network::NetworkController, 172

addUser
 guiMain::GuiMainController, 121
 guiMain::GuiMainInterface, 126
 guiMain::controller::menuController, 164

addUserToList
 data::DataController, 77

addUserToNetwork
 lo23::battleship::online::network::NetworkController, 172

addUserToUserList
 data::CDataCom, 34
 interfacesData::IDataCom, 139

address
 lo23::battleship::online::network::NetworkServer, 190

allBoatsArePlaced
 guiTable::controllers::PlacementPhaseController, 202

anchorPane
 guiTable::controllers::ObservationPhase, 197
 guiTable::controllers::PlacementPhaseController, 207
 guiTable::controllers::gameInterface, 102

askDisconnection
 data::CDataMain, 42
 interfacesData::IDataMain, 146
 lo23::battleship::online::network::COMInterface, 59
 lo23::battleship::online::network::NetworkModule↔Interface, 181

askJoinGame
 guiMain::GuiMainController, 121

attendedGame
 data::DataController, 84

avatar
 structData::Profile, 226

avatarImage
 guiMain::controller::SignupController, 245
 guiMain::controller::menuController, 167

avatarPath
 guiMain::controller::ChangeProfileController, 52
 guiMain::controller::SignupController, 246

BELGE_PLACEMENT_URL
 guiTable::controllers::GuiTableController, 132

BOATPHASE
 structData::StatusGame, 247

BOAT
 guiTable::CaseDrawing::Type, 247

backButton
 guiMain::controller::ChangeProfileController, 52
 guiMain::controller::ProfileController, 219

backToConnectionWindow
 guiMain::controller::SignupController, 245

- backToMenu
 - guiMain::controller::ChangeProfileController, 51
- backToTheMenu
 - guiMain::controller::ProfilController, 218
- backlog
 - lo23::battleship::online::network::NetworkServer, 190
- badPlacementColor
 - guiTable::BoatDrawing, 29
- belgiumGameType
 - guiMain::controller::CreateGameController, 73
- birthDatePicker
 - guiMain::controller::SignupController, 246
- birthdate
 - guiMain::controller::ChangeProfileController, 52
 - guiMain::controller::ProfilController, 219
 - structData::Profile, 226
- Boat
 - structData::Boat, 23, 24
- boat
 - lo23::battleship::online::network::messages::↔ ShotNotificationResultForSpectatorMessage, 241
 - lo23::battleship::online::network::messages::↔ ShotNotificationResultMessage, 243
- BoatDrawing
 - guiTable::BoatDrawing, 27
- boatMap
 - guiTable::controllers::PlacementPhaseController, 207
- boatRectangle
 - guiTable::BoatDrawing, 29
- BoatType
 - structData::BoatType, 30
- boatType
 - guiTable::BoatDrawing, 29
- boats
 - guiTable::controllers::GuiTableController, 132
- CDataCom
 - data::CDataCom, 34
- CDataMain
 - data::CDataMain, 42
- CDataTable
 - data::CDataTable, 47
- CHAT_FXML_URL
 - guiTable::controllers::GuiTableController, 132
- CLASSIC_PLACEMENT_URL
 - guiTable::controllers::GuiTableController, 132
- CONTRETORPILLEUR
 - structData::BoatType, 31
- CROISEURFR
 - structData::BoatType, 31
- CROISEURB
 - structData::BoatType, 31
- CUIRASSE
 - structData::BoatType, 31
- CaseDrawing
 - guiTable::CaseDrawing, 32
- changeProfileController
 - guiMain::GuiMainController, 125
- changeStatusGame
 - data::CDataCom, 35
 - interfacesData::IDataCom, 139
 - lo23::battleship::online::network::COMInterface, 59
 - lo23::battleship::online::network::NetworkModule↔ Interface, 182
- changeStatusGameStarted
 - data::CDataTable, 47
 - interfacesData::IDataTable, 151
- chatAutorisation
 - guiMain::controller::WaitingRoomController, 255
- chatAutorise
 - guiMain::controller::CreateGameController, 74
- chatController
 - guiTable::controllers::GuiTableController, 132
- ChatMessage
 - structData::ChatMessage, 55
- chatPane
 - guiTable::controllers::ObservationPhase, 197
 - guiTable::controllers::PlacementPhaseController, 207
 - guiTable::controllers::gameInterface, 102
- chatPane1
 - guiTable::controllers::ObservationPhase, 197
- chooseAvatar
 - guiMain::controller::SignupController, 245
- chooseAvatarButton
 - guiMain::controller::SignupController, 246
- classic
 - guiTable::controllers::GuiTableController, 132
- classicGameType
 - guiMain::controller::CreateGameController, 74
- classicType
 - structData::Game, 97
- cleanUp
 - structData::Profile, 222
- clear
 - data::CDataMain, 42
 - interfacesData::IDataMain, 146
- clearData
 - data::DataController, 77
- clearNetwork
 - lo23::battleship::online::network::COMInterface, 60
 - lo23::battleship::online::network::NetworkController, 172
 - lo23::battleship::online::network::NetworkModule↔ Interface, 182
- clone
 - structData::Game, 94
 - structData::Position, 215
 - structData::Profile, 222
- cloneBoat
 - structData::Boat, 24
- cloneChatMessage
 - structData::ChatMessage, 56
- cloneDataUser

- structData::DataUser, 87
- clonePlayer
 - structData::Player, 210
- cloneUser
 - structData::User, 251
- close
 - lo23::battleship::online::network::NetworkServer, 190
- closeListener
 - lo23::battleship::online::network::NetworkController, 172
- closeMsg
 - guiTable::controllers::PlacementPhaseController, 202
- closeSocket
 - lo23::battleship::online::network::NetworkListener, 178
- closeWindow
 - guiMain::controller::WaitingRoomController, 254
- compareProfileToPlayer
 - structData::Player, 210
- connection
 - data::CDataMain, 42
 - interfacesData::IDataMain, 146
- ConnectionEstablishedMessage
 - lo23::battleship::online::network::messages::← ConnectionEstablishedMessage, 65
- connectionPageButton
 - guiMain::controller::SignupController, 246
- ConnectionRequestMessage
 - lo23::battleship::online::network::messages::← ConnectionRequestMessage, 67
- ContactGroup
 - structData::ContactGroup, 69
- content
 - structData::ChatMessage, 57
- contreTorpilleurRectangle
 - guiTable::controllers::ClassicPlacementPhase← Controller, 58
- controller
 - data::CDataCom, 40
 - data::CDataMain, 46
 - data::CDataTable, 49
 - guiMain::GameCell, 100
 - guiMain::PlayerCell, 213
 - lo23::battleship::online::network::NetworkModule← Interface, 186
 - lo23::battleship::online::network::NetworkSender, 188
- conversation
 - guiTable::controllers::ChatController, 54
- conversationArea
 - guiTable::controllers::ChatController, 54
- coordinate
 - data::CDataTable, 47
 - interfacesData::IDataTable, 151
- coordinateShips
 - data::CDataTable, 48
- interfacesData::IDataTable, 151
- coordinates
 - data::CDataCom, 35
 - interfacesData::IDataCom, 139
 - lo23::battleship::online::network::COMInterface, 60
 - lo23::battleship::online::network::NetworkModule← Interface, 182
- createAccount
 - data::CDataMain, 42
 - guiMain::controller::LoginController, 161
 - guiMain::controller::SignupController, 245
 - interfacesData::IDataMain, 146
- createButton
 - guiMain::controller::SignupController, 246
- createConversation
 - guiTable::controllers::ChatController, 53
- createGame
 - guiMain::controller::CreateGameController, 74
- createGameController
 - guiMain::GuiMainController, 125
- createdGame
 - lo23::battleship::online::network::messages::← ConnectionEstablishedMessage, 66
- CreatedGameNotificationMessage
 - lo23::battleship::online::network::messages::← CreatedGameNotificationMessage, 70
- croiseurRectangle
 - guiTable::controllers::ClassicPlacementPhase← Controller, 58
- croiseurRectangle1
 - guiTable::controllers::BelgianPlacementPhase← Controller, 22
- croiseurRectangle2
 - guiTable::controllers::BelgianPlacementPhase← Controller, 22
- cuirasseRectangle
 - guiTable::controllers::BelgianPlacementPhase← Controller, 22
- DEFAULT_PORT
 - structData::User, 253
- DEFEAT_MSG
 - guiTable::controllers::gameInterface, 102
- data, 13
 - data.CDataCom, 32
 - data.CDataMain, 41
 - data.CDataTable, 46
 - data.DataController, 74
 - data::CDataCom
 - addNewGameList, 34
 - addUserToUserList, 34
 - CDataCom, 34
 - changeStatusGame, 35
 - controller, 40
 - coordinates, 35
 - getCreatedGame, 35
 - getLocalUser, 35
 - getOtherPlayer, 36
 - getPlayerPosition, 36

- getUserProfile, 36
- interfaceCom, 40
- interfaceMain, 40
- interfaceTable, 40
- joinGameSpectator, 36
- newRequestSpectator, 37
- notifToJoinGame, 37
- notifiedGameWon, 37
- notifyAttendedGameWon, 37
- notifyQuitSpectator, 37
- notifyToSpecGame, 38
- receiveMessage, 38
- receiveReady, 38
- removeGame, 38
- removeGameFromList, 38
- removeUser, 39
- sendStatistics, 39
- setGameJoinResponse, 39
- setInterfaceCom, 40
- setInterfaceMain, 40
- setInterfaceTable, 40
- setLocalGame, 40
- updateAttendedGame, 40
- data::CDataMain
 - askDisconnection, 42
 - CDataMain, 42
 - clear, 42
 - connection, 42
 - controller, 46
 - createAccount, 42
 - editProfile, 43
 - gameToSpec, 43
 - getGames, 43
 - getListUsers, 44
 - getLocalProfile, 44
 - getProfile, 44
 - interfaceCom, 46
 - newGame, 44
 - notifGameChosen, 45
 - removeGame, 45
 - setInterfaceCom, 45
 - setListIps, 45
 - setLocalGame, 45
 - setPort, 46
- data::CDataTable
 - CDataTable, 47
 - changeStatusGameStarted, 47
 - controller, 49
 - coordinate, 47
 - coordinateShips, 48
 - exit, 48
 - gameEnded, 48
 - getLocalGame, 48
 - getObserverGame, 48
 - interfaceCom, 49
 - interfaceMain, 49
 - interfaceTable, 49
 - NB_CASES_GRID, 50
 - setInterfaceCom, 48
 - setInterfaceMain, 49
 - setInterfaceTable, 49
 - textMessage, 49
 - timerOver, 49
- data::DataController
 - addGameToList, 77
 - addUserToList, 77
 - attendedGame, 84
 - clearData, 77
 - DataController, 77
 - endGame, 77
 - gameOver, 77
 - getAttendedGame, 78
 - getInterfaceDataCom, 78
 - getInterfaceDataMain, 78
 - getInterfaceDataTable, 78
 - getListGames, 78
 - getListUsers, 78
 - getLocalDataUser, 79
 - getLocalGame, 79
 - getLocalPlayer, 79
 - getLocalPlayerInGame, 79
 - getLocalProfile, 79
 - getLocalUser, 79
 - getOtherPLayer, 80
 - getPlayerPosition, 80
 - getTableInterface, 80
 - immediateDefeat, 80
 - interfaceCom, 84
 - interfaceDataCom, 85
 - interfaceDataMain, 85
 - interfaceDataTable, 85
 - interfaceMain, 85
 - interfaceTable, 85
 - isPlayer1, 80
 - isPlayerOf, 80
 - listGames, 85
 - listUsers, 85
 - localGame, 85
 - localPlayer, 85
 - localProfile, 85
 - localUser, 85
 - recordVictory, 81
 - reloadSavedProfile, 81
 - removeGameFromList, 81
 - removeUserFromList, 81
 - setAttendedGame, 81
 - setInterfaceCom, 82
 - setInterfaceMain, 82
 - setInterfaceTable, 82
 - setListGame, 82
 - setListUser, 82
 - setLocalDataUser, 83
 - setLocalGame, 83
 - setLocalPlayer, 83
 - setLocalProfile, 83
 - setLocalUser, 83

- testShot, [84](#)
- updateGameData, [84](#)
- updateGameDataPlaying, [84](#)
- updateGameStatus, [84](#)
- DataController
 - data::DataController, [77](#)
- dataController
 - guiTable::controllers::ChatController, [54](#)
 - guiTable::controllers::GuiTableController, [133](#)
- dataInterface
 - lo23::battleship::online::network::NetworkController, [176](#)
 - lo23::battleship::online::network::NetworkListener, [179](#)
 - lo23::battleship::online::network::NetworkModule←Interface, [186](#)
 - lo23::battleship::online::network::NetworkServer, [190](#)
- DataUser
 - structData::DataUser, [86](#), [87](#)
- desactiveBoat
 - guiTable::controllers::PlacementPhaseController, [202](#)
- destUser
 - lo23::battleship::online::network::messages::←NotifyReadyMessage, [193](#)
- disableRotation
 - guiTable::controllers::PlacementPhaseController, [202](#)
- disactiveColor
 - guiTable::BoatDrawing, [29](#)
- disconnection
 - guiMain::controller::menuController, [164](#)
- DisconnectionMessage
 - lo23::battleship::online::network::messages::←DisconnectionMessage, [89](#)
- displayDefeat
 - guiTable::GuiTableInterface, [134](#)
 - guiTable::controllers::GuiTableController, [130](#)
- displayMyShotResult
 - guiTable::GuiTableInterface, [134](#)
 - guiTable::controllers::GuiTableController, [130](#)
- displayObserverPhase
 - guiTable::GuiTableInterface, [135](#)
 - guiTable::controllers::GuiTableController, [130](#)
- displayObserverPhaseVictory
 - guiTable::GuiTableInterface, [135](#)
 - guiTable::controllers::GuiTableController, [130](#)
- displayObserverShot
 - guiTable::GuiTableInterface, [135](#)
 - guiTable::controllers::GuiTableController, [131](#)
- displayOpponentShot
 - guiTable::GuiTableInterface, [135](#)
 - guiTable::controllers::GuiTableController, [131](#)
- displayPlacementPhase
 - guiTable::GuiTableInterface, [136](#)
 - guiTable::controllers::GuiTableController, [131](#)
- displayRageQuit
 - guiTable::GuiTableInterface, [136](#)
 - guiTable::controllers::GuiTableController, [131](#)
- displayShot
 - guiTable::controllers::ObservationPhase, [195](#)
 - guiTable::controllers::ObserverPhaseController, [199](#)
- displayVictory
 - guiTable::GuiTableInterface, [136](#)
 - guiTable::controllers::GuiTableController, [131](#)
- doProfileArea
 - guiTable::controllers::ChatController, [54](#)
- doesProfileBelongToGame
 - structData::Game, [94](#)
- draw
 - guiTable::controllers::PlacementPhaseController, [203](#)
- drawBoatsNewPosition
 - guiTable::controllers::PlacementPhaseController, [204](#)
- drawRotation
 - guiTable::controllers::PlacementPhaseController, [204](#)
- ERROR
 - structData::MessageType, [170](#)
- EXIT_GAME_MSG
 - guiTable::controllers::ObservationPhase, [197](#)
 - guiTable::controllers::gameInterface, [102](#)
- EXPLAIN_PLACEMENT
 - guiTable::controllers::PlacementPhaseController, [207](#)
- editProfile
 - data::CDataMain, [43](#)
 - interfacesData::IDataMain, [147](#)
- enableRotation
 - guiTable::controllers::PlacementPhaseController, [204](#)
- endGame
 - data::DataController, [77](#)
- errorMessage
 - guiMain::controller::ChangeProfileController, [52](#)
 - guiMain::controller::CreateGameController, [74](#)
 - guiMain::controller::LoginController, [161](#)
- exit
 - data::CDataTable, [48](#)
 - interfacesData::IDataTable, [151](#)
- exitButton
 - guiTable::controllers::ObservationPhase, [198](#)
 - guiTable::controllers::gameInterface, [102](#)
- exitGame
 - guiTable::controllers::GamePhaseController, [105](#)
 - guiTable::controllers::GuiTableController, [131](#)
 - guiTable::controllers::ObservationPhase, [195](#)
 - guiTable::controllers::ObserverPhaseController, [199](#)
- FINISHED
 - structData::StatusGame, [247](#)
- field

- guiTable::controllers::ChatController, 54
- fillChatSlot
 - guiTable::controllers::BaseController, 19
- fillElement
 - guiTable::controllers::BaseController, 20
- filterKnownIPAddressesToNotify
 - lo23::battleship::online::network::NetworkController, 172
- filterUnknownIPAddresses
 - lo23::battleship::online::network::NetworkController, 173
- firstName
 - guiMain::controller::ChangeProfileController, 52
 - guiMain::controller::ProfilController, 219
- firstNameTextField
 - guiMain::controller::SignupController, 246
- GAME_PHASE_URL
 - guiTable::controllers::GuiTableController, 133
- GRID_ELEMENT_SIZE
 - guiTable::controllers::BaseController, 20
- GRID_X
 - guiTable::controllers::BaseController, 20
- GRID_Y
 - guiTable::controllers::BaseController, 20
- Game
 - structData::Game, 92, 93
- game
 - lo23::battleship::online::network::messages::↔ ConnectionRequestMessage, 68
 - lo23::battleship::online::network::messages::↔ CreatedGameNotificationMessage, 71
 - lo23::battleship::online::network::messages::↔ DisconnectionMessage, 90
 - lo23::battleship::online::network::messages::↔ GameQuitMessage, 109
 - lo23::battleship::online::network::messages::↔ GameQuitSpectatorMessage, 111
 - lo23::battleship::online::network::messages::Join↔ GameRequestMessage, 157
 - lo23::battleship::online::network::messages::Join↔ GameResponseMessage, 160
 - lo23::battleship::online::network::messages::↔ SendInfoGameForSpectatorMessage, 231
- GameCell
 - guiMain::GameCell, 99
- gameEnded
 - data::CDataTable, 48
 - interfacesData::IDataTable, 151
- gameName
 - guiMain::controller::CreateGameController, 74
 - guiMain::controller::WaitingRoomController, 255
- gameOver
 - data::DataController, 77
- gamePhaseController
 - guiTable::controllers::GuiTableController, 133
- GameQuitMessage
 - lo23::battleship::online::network::messages::↔ GameQuitMessage, 109
- gameQuitSpectator
 - lo23::battleship::online::network::COMInterface, 60
 - lo23::battleship::online::network::NetworkModule↔ Interface, 182
- GameQuitSpectatorMessage
 - lo23::battleship::online::network::messages::↔ GameQuitSpectatorMessage, 110
- gameState
 - guiTable::controllers::GamePhaseController, 107
 - guiTable::controllers::ObservationPhase, 198
 - guiTable::controllers::ObserverPhaseController, 200
- gameToSpec
 - data::CDataMain, 43
 - interfacesData::IDataMain, 147
- gameType
 - guiMain::controller::CreateGameController, 74
 - guiMain::controller::WaitingRoomController, 255
- gameUpdate
 - lo23::battleship::online::network::messages::↔ UpdateGameMessage, 249
- GameWonMessage
 - lo23::battleship::online::network::messages::↔ GameWonMessage, 112
- GameWonMessageToSpectator
 - lo23::battleship::online::network::messages::↔ GameWonMessageToSpectator, 114
- gamesAborted
 - structData::Profile, 226
- gamesList
 - guiMain::GuiMainController, 125
- gamesLost
 - structData::Profile, 226
- gamesPlayed
 - structData::Profile, 226
- gamesView
 - guiMain::controller::menuController, 167
- gamesWon
 - structData::Profile, 226
- getActiveColor
 - guiTable::BoatDrawing, 28
- getAddressForUser
 - lo23::battleship::online::network::NetworkController, 173
- getAttendedGame
 - data::DataController, 78
- getAvatar
 - structData::Profile, 223
- getBadPlacementColor
 - guiTable::BoatDrawing, 28
- getBirthdate
 - structData::Profile, 223
- getBoatRectangle
 - guiTable::BoatDrawing, 28
- getBoatSize
 - guiTable::BoatDrawing, 28
- getBoatType
 - guiTable::BoatDrawing, 28

- getCOMInterface
 - lo23::battleship::online::network::NetworkController, 173
- getChatPane
 - guiTable::controllers::GamePhaseController, 105
 - guiTable::controllers::ObservationPhase, 195
 - guiTable::controllers::PlacementPhaseController, 204
- getClassicType
 - structData::Game, 94
- getConnectedUsers
 - lo23::battleship::online::network::NetworkController, 173
- getContent
 - structData::ChatMessage, 56
- getConversation
 - guiTable::controllers::ChatController, 54
- getCreatedGame
 - data::CDataCom, 35
 - interfacesData::IDataCom, 140
- getDataController
 - guiTable::controllers::GuiTableController, 132
- getDisactiveColor
 - guiTable::BoatDrawing, 28
- getGames
 - data::CDataMain, 43
 - interfacesData::IDataMain, 147
- getGamesAborted
 - structData::Profile, 223
- getGamesLost
 - structData::Profile, 223
- getGamesPlayed
 - structData::Profile, 223
- getGamesWon
 - structData::Profile, 223
- getGridCol
 - guiTable::BoatDrawing, 28
- getGridRow
 - guiTable::BoatDrawing, 28
- getHumanOpponent
 - structData::Game, 94
- getIPTable
 - lo23::battleship::online::network::NetworkController, 174
- getIPs
 - structData::User, 252
- getIdGame
 - structData::Game, 94
- getIdUser
 - structData::User, 252
- getIdata
 - guiMain::GuiMainController, 121
- getImage
 - structData::Profile, 224
- getInfoGameForSpectator
 - lo23::battleship::online::network::COMInterface, 60
 - lo23::battleship::online::network::NetworkModule↔Interface, 183
- GetInfoGameForSpectatorMessage
 - lo23::battleship::online::network::messages::Get↔InfoGameForSpectatorMessage, 116
- getInitialLayoutX
 - guiTable::BoatDrawing, 28
- getInitialLayoutY
 - guiTable::BoatDrawing, 28
- getInstance
 - guiTable::controllers::GuiTableController, 132
 - lo23::battleship::online::network::NetworkController, 174
- getInterfaceDataCom
 - data::DataController, 78
- getInterfaceDataMain
 - data::DataController, 78
- getInterfaceDataTable
 - data::DataController, 78
- getIpAddress
 - lo23::battleship::online::network::NetworkServer, 190
- getIps
 - guiMain::GuiMainController, 121
- getIsRunning
 - lo23::battleship::online::network::NetworkListener, 178
- getLastname
 - structData::Profile, 224
- getLblj1
 - guiTable::controllers::ObservationPhase, 195
- getLblj2
 - guiTable::controllers::ObservationPhase, 195
- getListBoats
 - structData::Player, 210
- getListCases
 - structData::Boat, 24
- getListContacts
 - structData::DataUser, 87
- getListGames
 - data::DataController, 78
- getListMessages
 - structData::Game, 95
- getListShots
 - structData::Player, 210
- getListSpectators
 - structData::Game, 95
- getListUsers
 - data::CDataMain, 44
 - data::DataController, 78
 - interfacesData::IDataMain, 147
- getLocalDataUser
 - data::DataController, 79
- getLocalGame
 - data::CDataTable, 48
 - data::DataController, 79
 - interfacesData::IDataTable, 151
- getLocalPlayer
 - data::DataController, 79
- getLocalPlayerInGame

- data::DataController, 79
- getLocalProfile
 - data::CDataMain, 44
 - data::DataController, 79
 - interfacesData::IDataMain, 148
- getLocalUser
 - data::CDataCom, 35
 - data::DataController, 79
 - interfacesData::IDataCom, 140
- getLogin
 - structData::User, 252
- getName
 - structData::BoatType, 30
 - structData::Game, 95
 - structData::Profile, 224
- getNbCases
 - structData::BoatType, 30
- getObserverGame
 - data::CDataTable, 48
 - interfacesData::IDataTable, 152
- getOtherPPlayer
 - data::DataController, 80
- getOtherPlayer
 - data::CDataCom, 36
 - interfacesData::IDataCom, 140
- getPassword
 - structData::DataUser, 87
- getPlayer1
 - structData::Game, 95
- getPlayer1 Start
 - structData::Game, 95
- getPlayer2
 - structData::Game, 95
- getPlayerPosition
 - data::CDataCom, 36
 - data::DataController, 80
 - interfacesData::IDataCom, 140
- getPort
 - guiMain::GuiMainController, 121
 - lo23::battleship::online::network::NetworkController, 174
 - structData::User, 252
- getProfile
 - data::CDataMain, 44
 - interfacesData::IDataMain, 148
 - lo23::battleship::online::network::COMInterface, 61
 - lo23::battleship::online::network::NetworkModuleInterface, 183
 - structData::ChatMessage, 56
 - structData::Player, 211
- GetProfileRequestMessage
 - lo23::battleship::online::network::messages::Get←ProfileRequestMessage, 117
- getServerSocketIPAddress
 - lo23::battleship::online::network::NetworkListener, 178
- getSpectator
 - structData::Game, 96
- getSpectatorChat
 - structData::Game, 96
- getStatus
 - structData::Game, 96
- getSunk
 - structData::Boat, 24
- getTableInterface
 - data::DataController, 80
- getTime
 - structData::ChatMessage, 56
 - structData::Shot, 236
- getTimePerShot
 - structData::Game, 96
- getTimeToPlaceBoats
 - structData::Game, 96
- getTouched
 - structData::Position, 216
- getType
 - lo23::battleship::online::network::messages::←ConnectionEstablishedMessage, 65
 - lo23::battleship::online::network::messages::←ConnectionRequestMessage, 67
 - lo23::battleship::online::network::messages::←CreatedGameNotificationMessage, 70
 - lo23::battleship::online::network::messages::←DisconnectionMessage, 89
 - lo23::battleship::online::network::messages::←GameQuitMessage, 109
 - lo23::battleship::online::network::messages::←GameQuitSpectatorMessage, 110
 - lo23::battleship::online::network::messages::←GameWonMessage, 112
 - lo23::battleship::online::network::messages::←GameWonMessageToSpectator, 114
 - lo23::battleship::online::network::messages::Get←InfoGameForSpectatorMessage, 116
 - lo23::battleship::online::network::messages::Get←ProfileRequestMessage, 118
 - lo23::battleship::online::network::messages::Join←GameRequestMessage, 157
 - lo23::battleship::online::network::messages::Join←GameResponseMessage, 159
 - lo23::battleship::online::network::messages::←Message, 169
 - lo23::battleship::online::network::messages::←NotifyReadyMessage, 192
 - lo23::battleship::online::network::messages::←ProfileRequestAnswerMessage, 228
 - lo23::battleship::online::network::messages::←SendInfoGameForSpectatorMessage, 230
 - lo23::battleship::online::network::messages::←SendNewSpectatorMessage, 232
 - lo23::battleship::online::network::messages::←SendTextMessage, 234
 - lo23::battleship::online::network::messages::←ShotNotificationMessage, 238
 - lo23::battleship::online::network::messages::←ShotNotificationResultForSpectatorMessage,

- 241
- lo23::battleship::online::network::messages::↔
 - ShotNotificationResultMessage, 243
- lo23::battleship::online::network::messages::↔
 - UpdateGameMessage, 249
- structData::Boat, 25
- getUserProfile
 - data::CDataCom, 36
 - interfacesData::IDataCom, 140
- getUsername
 - structData::User, 252
- getX
 - structData::Position, 216
- getY
 - structData::Position, 216
- gridCol
 - guiTable::BoatDrawing, 29
- gridRow
 - guiTable::BoatDrawing, 29
- guiMain, 13
- guiMain.controller, 14
- guiMain.controller.ChangeProfileController, 50
- guiMain.controller.CreateGameController, 71
- guiMain.controller.IpConfigController, 154
- guiMain.controller.LoginController, 160
- guiMain.controller.menuController, 162
- guiMain.controller.ProfilController, 217
- guiMain.controller.SignupController, 244
- guiMain.controller.WaitingRoomController, 254
- guiMain.GameCell, 98
- guiMain.GuiMainController, 119
- guiMain.GuiMainInterface, 126
- guiMain.IpCell, 152
- guiMain.PlayerCell, 212
- guiMain::GameCell
 - controller, 100
 - GameCell, 99
 - hbox, 100
 - joinButton, 100
 - joinButtonBox, 100
 - lookButton, 100
 - lookButtonBox, 100
 - robot, 100
 - title, 100
 - type, 100
 - updateItem, 99
- guiMain::GuiMainController
 - addGame, 121
 - addUser, 121
 - askJoinGame, 121
 - changeProfileController, 125
 - createGameController, 125
 - gamesList, 125
 - getIdata, 121
 - getIps, 121
 - getPort, 121
 - GuiMainController, 120
 - idata, 125
 - ipConfigController, 125
 - ipsList, 125
 - loginController, 125
 - lookGame, 122
 - menuController, 125
 - openChangeProfileWindow, 122
 - openConfigWindow, 122
 - openCreateGameWindow, 122
 - openMenuWindow, 122
 - openPlacementPhase, 122
 - openProfileWindow, 122
 - openSignupWindow, 123
 - openWaitingRoomWindow, 123
 - playersList, 125
 - port, 125
 - profilController, 125
 - removeGame, 123
 - removeUser, 123
 - rootLayout, 125
 - sendStatistics, 123
 - setGameJoinResponse, 123
 - setIdata, 124
 - setIps, 124
 - setPort, 124
 - signupController, 125
 - stage, 125
 - startIHM, 124
 - transmitNewStatus, 124
 - waitingRoomController, 125
- guiMain::GuiMainInterface
 - addGame, 126
 - addUser, 126
 - openMenuWindow, 127
 - openPlacementPhase, 127
 - removeGame, 127
 - removeUser, 127
 - sendStatistics, 127
 - setGameJoinResponse, 127
 - transmitNewStatus, 128
- guiMain::IpCell
 - hbox, 153
 - IpCell, 153
 - ipLabel, 154
 - pane, 154
 - removeButton, 154
 - updateItem, 153
- guiMain::PlayerCell
 - controller, 213
 - hbox, 213
 - lookButton, 214
 - lookButtonBox, 214
 - name, 214
 - PlayerCell, 213
 - updateItem, 213
- guiMain::controller::ChangeProfileController
 - avatarPath, 52
 - backButton, 52
 - backToMenu, 51

- birthdate, 52
- errorMessage, 52
- firstName, 52
- init, 51
- initialize, 51
- lastName, 52
- mainController, 52
- modifyAvatar, 51
- modifyAvatarButton, 52
- nameTitle, 52
- numberOfGame, 52
- numberOfGameLost, 53
- numberOfGameWon, 53
- openFile, 51
- saveButton, 53
- saveProfile, 52
- setMainController, 52
- userAvatar, 53
- userName, 53
- userPassword, 53
- guiMain::controller::CreateGameController
 - belgiumGameType, 73
 - chatAutorise, 74
 - classicGameType, 74
 - createGame, 74
 - errorMessage, 74
 - gameName, 74
 - gameType, 74
 - initialize, 72
 - mainController, 74
 - newGame, 73
 - positioningTime, 74
 - reflectionTime, 74
 - returnButton, 74
 - returnToMenu, 73
 - setMainController, 73
 - spectatorsAutorise, 74
 - timeDisabled, 74
 - timerDisable, 73
- guiMain::controller::IpConfigController
 - addButton, 155
 - addIp, 155
 - init, 155
 - initIpsList, 155
 - ipTextField, 155
 - ipsListView, 155
 - mainController, 155
 - port, 155
 - setMainController, 155
 - validate, 155
 - validateButton, 155
- guiMain::controller::LoginController
 - createAccount, 161
 - errorMessage, 161
 - initialize, 160
 - login, 160
 - loginButton, 161
 - mainController, 161
 - openCreateAccountWindow, 161
 - setMainController, 161
 - userLogin, 161
 - userPassword, 161
- guiMain::controller::ProfilController
 - backButton, 219
 - backToTheMenu, 218
 - birthdate, 219
 - firstName, 219
 - init, 218
 - initialize, 218
 - lastName, 219
 - mainController, 219
 - nameTitle, 219
 - numberOfGame, 219
 - numberOfGameLost, 219
 - numberOfGameWon, 219
 - setMainController, 219
 - setProfil, 219
 - userAvatar, 220
 - userName, 220
- guiMain::controller::SignupController
 - addConnectionButton, 245
 - addConnectionPoint, 245
 - avatarImage, 245
 - avatarPath, 246
 - backToConnectionWindow, 245
 - birthDatePicker, 246
 - chooseAvatar, 245
 - chooseAvatarButton, 246
 - connectionPageButton, 246
 - createAccount, 245
 - createButton, 246
 - firstNameTextField, 246
 - idTextField, 246
 - init, 245
 - lastNameTextField, 246
 - mainController, 246
 - messageLabel, 246
 - openFile, 245
 - passwordTextField, 246
 - setMainController, 245
 - usernameTextField, 246
- guiMain::controller::WaitingRoomController
 - chatAutorisation, 255
 - closeWindow, 254
 - gameName, 255
 - gameType, 255
 - initData, 254
 - initialize, 255
 - reflectionTime, 255
 - setStage, 255
 - spectatorAutorisation, 255
 - stage, 255
- guiMain::controller::menuController
 - addGame, 164
 - addUser, 164
 - avatarImage, 167

- disconnection, 164
- gamesView, 167
- init, 164
- initGamesList, 164
- initUserList, 165
- initialize, 165
- joinGame, 165
- lookGame, 165
- lookUser, 165
- mainController, 167
- modifyProfileButton, 167
- openChangeProfileWindow, 165
- openCreateGameWindow, 166
- option, 166
- optionButton, 167
- playersView, 167
- refresh, 166
- refreshButton, 167
- removeGame, 166
- removeUser, 166
- setImage, 166
- setMainController, 167
- updateGameStatus, 167
- warningLabel, 167
- GuiMainController
 - guiMain::GuiMainController, 120
- guiTable, 14
 - guiTable.BoatDrawing, 26
 - guiTable.CaseDrawing, 31
 - guiTable.CaseDrawing.Type, 247
 - guiTable.controllers, 14
 - guiTable.controllers.BaseController, 19
 - guiTable.controllers.BelgianPlacementPhaseController, 21
 - guiTable.controllers.ChatController, 53
 - guiTable.controllers.ClassicPlacementPhaseController, 57
 - guiTable.controllers.gameInterface, 100
 - guiTable.controllers.GamePhaseController, 103
 - guiTable.controllers.GuiTableController, 128
 - guiTable.controllers.ObservationPhase, 193
 - guiTable.controllers.ObserverPhaseController, 199
 - guiTable.controllers.PlacementPhaseController, 200
 - guiTable.controllers.ProfileController, 227
 - guiTable.GuiTableInterface, 133
 - guiTable::BoatDrawing
 - active, 29
 - activeColor, 29
 - badPlacementColor, 29
 - BoatDrawing, 27
 - boatRectangle, 29
 - boatType, 29
 - disactiveColor, 29
 - getActiveColor, 28
 - getBadPlacementColor, 28
 - getBoatRectangle, 28
 - getBoatSize, 28
 - getBoatType, 28
 - getDisactiveColor, 28
 - getGridCol, 28
 - getGridRow, 28
 - getInitialLayoutX, 28
 - getInitialLayoutY, 28
 - gridCol, 29
 - gridRow, 29
 - initialLayoutX, 29
 - initialLayoutY, 29
 - isActive, 28
 - isPlaced, 28
 - isRotation, 28
 - placed, 29
 - reinit, 28
 - rotation, 29
 - setActive, 28
 - setActiveBoat, 28
 - setBoatRectangle, 29
 - setBoatType, 29
 - setPlaced, 29
 - setPosition, 29
 - setRotation, 29
 - guiTable::CaseDrawing
 - CaseDrawing, 32
 - guiTable::CaseDrawing::Type
 - BOAT, 247
 - MISSED, 247
 - SHOT, 247
 - SUNK_BOAT, 247
 - TOUCHED, 247
 - guiTable::GuiTableInterface
 - addChatMessage, 134
 - displayDefeat, 134
 - displayMyShotResult, 134
 - displayObserverPhase, 135
 - displayObserverPhaseVictory, 135
 - displayObserverShot, 135
 - displayOpponentShot, 135
 - displayPlacementPhase, 136
 - displayRageQuit, 136
 - displayVictory, 136
 - opponentReady, 136
 - setDataController, 137
 - sunkPlayerBoat, 137
 - updateSpectatorGame, 137
 - guiTable::controllers::BaseController
 - fillChatSlot, 19
 - fillElement, 20
 - GRID_ELEMENT_SIZE, 20
 - GRID_X, 20
 - GRID_Y, 20
 - NB_CASES_GRID, 20
 - SPACE, 20
 - guiTable::controllers::BelgianPlacementPhaseController
 - croiseurRectangle1, 22
 - croiseurRectangle2, 22
 - cuirasseRectangle, 22
 - initBoatMap, 22

- sousMarinRectangle1, [22](#)
- sousMarinRectangle2, [22](#)
- sousMarinRectangle3, [22](#)
- sousMarinRectangle4, [22](#)
- torpilleurRectangle1, [22](#)
- torpilleurRectangle2, [22](#)
- torpilleurRectangle3, [22](#)
- guiTable::controllers::ChatController
 - conversation, [54](#)
 - conversationArea, [54](#)
 - createConversation, [53](#)
 - dataController, [54](#)
 - doProfileArea, [54](#)
 - field, [54](#)
 - getConversation, [54](#)
 - init, [54](#)
 - profils, [54](#)
 - receiveAMessage, [54](#)
 - reloadConversation, [54](#)
 - sendMyMessage, [54](#)
 - setDataController, [54](#)
- guiTable::controllers::ClassicPlacementPhaseController
 - contreTorpilleurRectangle, [58](#)
 - croiseurRectangle, [58](#)
 - initBoatMap, [58](#)
 - porteAvionsRectangle, [58](#)
 - sousMarinRectangle, [58](#)
 - torpilleurRectangle, [58](#)
- guiTable::controllers::GamePhaseController
 - addOpponentShot, [105](#)
 - addShot, [105](#)
 - exitGame, [105](#)
 - gameState, [107](#)
 - getChatPane, [105](#)
 - initialize, [105](#)
 - messageMask, [107](#)
 - myTable, [107](#)
 - myTurn, [107](#)
 - noClicked, [105](#)
 - onClickCase, [105](#)
 - removeSelectedCase, [106](#)
 - selectedCase, [107](#)
 - setMyBoats, [106](#)
 - setMyTurn, [106](#)
 - setRoundTime, [106](#)
 - showDefeat, [106](#)
 - showVictory, [106](#)
 - sunckBoat, [106](#)
 - sunkMyBoat, [107](#)
 - table, [107](#)
 - time, [107](#)
 - timeIsOver, [107](#)
 - timePerShot, [107](#)
 - timeline, [107](#)
 - timerLabel, [107](#)
 - validateShot, [107](#)
 - valider, [107](#)
- guiTable::controllers::GuiTableController
 - addChatMessage, [130](#)
 - BELGE_PLACEMENT_URL, [132](#)
 - boats, [132](#)
 - CHAT_FXML_URL, [132](#)
 - CLASSIC_PLACEMENT_URL, [132](#)
 - chatController, [132](#)
 - classic, [132](#)
 - dataController, [133](#)
 - displayDefeat, [130](#)
 - displayMyShotResult, [130](#)
 - displayObserverPhase, [130](#)
 - displayObserverPhaseVictory, [130](#)
 - displayObserverShot, [131](#)
 - displayOpponentShot, [131](#)
 - displayPlacementPhase, [131](#)
 - displayRageQuit, [131](#)
 - displayVictory, [131](#)
 - exitGame, [131](#)
 - GAME_PHASE_URL, [133](#)
 - gamePhaseController, [133](#)
 - getDataController, [132](#)
 - getInstance, [132](#)
 - GuiTableController, [130](#)
 - INSTANCE, [133](#)
 - mainStage, [133](#)
 - OBSERVATION_PHASE_URL, [133](#)
 - OBSERVER_PHASE_URL, [133](#)
 - observationControlleur, [133](#)
 - observerPhaseController, [133](#)
 - opponentReady, [132](#)
 - placementPhaseController, [133](#)
 - rootLayout, [133](#)
 - setDataController, [132](#)
 - sunkPlayerBoat, [132](#)
 - TITLE, [133](#)
 - updateSpectatorGame, [132](#)
 - validateBoats, [132](#)
 - validateShot, [132](#)
- guiTable::controllers::ObservationPhase
 - anchorPane, [197](#)
 - chatPane, [197](#)
 - chatPane1, [197](#)
 - displayShot, [195](#)
 - EXIT_GAME_MSG, [197](#)
 - exitButton, [198](#)
 - exitGame, [195](#)
 - gameState, [198](#)
 - getChatPane, [195](#)
 - getLblj1, [195](#)
 - getLblj2, [195](#)
 - init, [195](#)
 - lblj1, [198](#)
 - lblj2, [198](#)
 - logMsg, [195](#)
 - logYesNoMsg, [196](#)
 - MY_TURN_MSG, [198](#)
 - messageContainer, [198](#)
 - messageTextContainer, [198](#)

- noButton, 198
- noClicked, 196
- OTHER_TURN_MSG, 198
- placeShotTo, 196
- STYLE_MY_TURN, 198
- STYLE_OTHER_TURN, 198
- setLblj1, 196
- setLblj2, 196
- setTableController, 196
- setTurn, 196
- showVictory, 197
- sunkABoat, 197
- sunkPlayerBoat, 197
- table1, 198
- table2, 198
- tableController, 198
- tablePlayer, 198
- timerLabel, 198
- turn, 198
- VICTORY_P1_MSG, 198
- VICTORY_P2_MSG, 198
- waitExit, 198
- waitMsg, 198
- yesButton, 198
- yesClicked, 197
- guiTable::controllers::ObserverPhaseController
 - displayShot, 199
 - exitGame, 199
 - gameState, 200
 - initialize, 199
 - noClicked, 199
 - setTurn, 199
 - sunkPlayerBoat, 200
 - table1, 200
 - table2, 200
 - tablePlayer, 200
- guiTable::controllers::PlacementPhaseController
 - activateBoat, 202
 - activeBoat, 207
 - allBoatsArePlaced, 202
 - anchorPane, 207
 - boatMap, 207
 - chatPane, 207
 - closeMsg, 202
 - desactiveBoat, 202
 - disableRotation, 202
 - draw, 203
 - drawBoatsNewPosition, 204
 - drawRotation, 204
 - EXPLAIN_PLACEMENT, 207
 - enableRotation, 204
 - getChatPane, 204
 - initBoatMap, 204
 - initialize, 204
 - isIsValidate, 205
 - isValidate, 207
 - logMsg, 205
 - messageContainer, 207
 - MousePlaceBoat, 205
 - onValidate, 205
 - placeBoat, 205
 - playKeyEvent, 205
 - positionCorrect, 206
 - RANDOM_ROTATION, 207
 - reinitBoat, 206
 - rotationIsValide, 207
 - setIsValidate, 206
 - setPlacementTime, 206
 - table, 207
 - time, 207
 - timelsOver, 206
 - timePlacement, 207
 - timeline, 207
 - timerLabel, 207
 - valider, 207
- guiTable::controllers::ProfileController
 - init, 227
- guiTable::controllers::gameInterface
 - anchorPane, 102
 - chatPane, 102
 - DEFEAT_MSG, 102
 - EXIT_GAME_MSG, 102
 - exitButton, 102
 - logMsg, 101
 - logYesNoMsg, 102
 - MY_TURN_MSG, 103
 - messageContainer, 102
 - messageTextContainer, 102
 - noButton, 103
 - OTHER_TURN_MSG, 103
 - placeShotTo, 102
 - STYLE_MY_TURN, 103
 - STYLE_OTHER_TURN, 103
 - showDefeat, 102
 - showVictory, 102
 - sunkABoat, 102
 - tableController, 103
 - VICTORY_MSG, 103
 - waitExit, 103
 - waitMsg, 103
 - yesButton, 103
 - yesClicked, 102
- GuiTableController
 - guiTable::controllers::GuiTableController, 130
- hbox
 - guiMain::GameCell, 100
 - guiMain::IpCell, 153
 - guiMain::PlayerCell, 213
- host
 - lo23::battleship::online::network::NetworkSender, 188
- humanOpponent
 - structData::Game, 97
- INSTANCE
 - guiTable::controllers::GuiTableController, 133

- iPs
 - structData::User, [253](#)
- idGame
 - structData::Game, [97](#)
- idTextField
 - guiMain::controller::SignupController, [246](#)
- idUser
 - structData::User, [253](#)
- idata
 - guiMain::GuiMainController, [125](#)
- immediateDefeat
 - data::DataController, [80](#)
- init
 - guiMain::controller::ChangeProfileController, [51](#)
 - guiMain::controller::IpConfigController, [155](#)
 - guiMain::controller::ProfilController, [218](#)
 - guiMain::controller::SignupController, [245](#)
 - guiMain::controller::menuController, [164](#)
 - guiTable::controllers::ChatController, [54](#)
 - guiTable::controllers::ObservationPhase, [195](#)
 - guiTable::controllers::ProfileController, [227](#)
- initBoatMap
 - guiTable::controllers::BelgianPlacementPhase←
Controller, [22](#)
 - guiTable::controllers::ClassicPlacementPhase←
Controller, [58](#)
 - guiTable::controllers::PlacementPhaseController,
[204](#)
- initData
 - guiMain::controller::WaitingRoomController, [254](#)
- initGamesList
 - guiMain::controller::menuController, [164](#)
- initIpsList
 - guiMain::controller::IpConfigController, [155](#)
- initUserList
 - guiMain::controller::menuController, [165](#)
- initialLayoutX
 - guiTable::BoatDrawing, [29](#)
- initialLayoutY
 - guiTable::BoatDrawing, [29](#)
- initialize
 - guiMain::controller::ChangeProfileController, [51](#)
 - guiMain::controller::CreateGameController, [72](#)
 - guiMain::controller::LoginController, [160](#)
 - guiMain::controller::ProfilController, [218](#)
 - guiMain::controller::WaitingRoomController, [255](#)
 - guiMain::controller::menuController, [165](#)
 - guiTable::controllers::GamePhaseController, [105](#)
 - guiTable::controllers::ObserverPhaseController,
[199](#)
 - guiTable::controllers::PlacementPhaseController,
[204](#)
- instance
 - lo23::battleship::online::network::NetworkController,
[176](#)
- interfaceCom
 - data::CDataCom, [40](#)
 - data::CDataMain, [46](#)
- data::CDataTable, [49](#)
- data::DataController, [84](#)
- interfaceDataCom
 - data::DataController, [85](#)
- interfaceDataMain
 - data::DataController, [85](#)
- interfaceDataTable
 - data::DataController, [85](#)
- interfaceMain
 - data::CDataCom, [40](#)
 - data::CDataTable, [49](#)
 - data::DataController, [85](#)
- interfaceTable
 - data::CDataCom, [40](#)
 - data::CDataTable, [49](#)
 - data::DataController, [85](#)
- interfacesData, [15](#)
- interfacesData.IDataCom, [137](#)
- interfacesData.IDataMain, [145](#)
- interfacesData.IDataTable, [150](#)
- interfacesData::IDataCom
 - addNewGameList, [138](#)
 - addUserToUserList, [139](#)
 - changeStatusGame, [139](#)
 - coordinates, [139](#)
 - getCreatedGame, [140](#)
 - getLocalUser, [140](#)
 - getOtherPlayer, [140](#)
 - getPlayerPosition, [140](#)
 - getUserProfile, [140](#)
 - joinGameSpectator, [141](#)
 - newRequestSpectator, [141](#)
 - notifToJoinGame, [141](#)
 - notifiedGameWon, [141](#)
 - notifyAttendedGameWon, [142](#)
 - notifyQuitSpectator, [142](#)
 - notifyToSpecGame, [142](#)
 - receiveMessage, [142](#)
 - receiveReady, [142](#)
 - removeGame, [143](#)
 - removeGameFromList, [143](#)
 - removeUser, [143](#)
 - sendStatistics, [143](#)
 - setGameJoinResponse, [143](#), [144](#)
 - setLocalGame, [144](#)
 - updateAttendedGame, [144](#)
- interfacesData::IDataMain
 - askDisconnection, [146](#)
 - clear, [146](#)
 - connection, [146](#)
 - createAccount, [146](#)
 - editProfile, [147](#)
 - gameToSpec, [147](#)
 - getGames, [147](#)
 - getListUsers, [147](#)
 - getLocalProfile, [148](#)
 - getProfile, [148](#)
 - newGame, [148](#)

- notifGameChosen, 148
 - removeGame, 149
 - setListIps, 149
 - setLocalGame, 149
 - setPort, 149
- interfacesData::IDataTable
 - changeStatusGameStarted, 151
 - coordinate, 151
 - coordinateShips, 151
 - exit, 151
 - gameEnded, 151
 - getLocalGame, 151
 - getObserverGame, 152
 - textMessage, 152
 - timerOver, 152
- ipAdressesTable
 - lo23::battleship::online::network::messages::↵
ConnectionEstablishedMessage, 66
 - lo23::battleship::online::network::messages::↵
ConnectionRequestMessage, 68
- IpCell
 - guiMain::IpCell, 153
- ipConfigController
 - guiMain::GuiMainController, 125
- ipLabel
 - guiMain::IpCell, 154
- ipTextField
 - guiMain::controller::IpConfigController, 155
- ipsList
 - guiMain::GuiMainController, 125
- ipsListView
 - guiMain::controller::IpConfigController, 155
- isActive
 - guiTable::BoatDrawing, 28
- isIsValidate
 - guiTable::controllers::PlacementPhaseController, 205
- isOk
 - lo23::battleship::online::network::messages::Join↵
GameResponseMessage, 160
- isOpened
 - lo23::battleship::online::network::NetworkServer, 190
- isPlaced
 - guiTable::BoatDrawing, 28
- isPlayer1
 - data::DataController, 80
- isPlayerOf
 - data::DataController, 80
- isReady
 - structData::Player, 211
- isRotation
 - guiTable::BoatDrawing, 28
- isRunning
 - lo23::battleship::online::network::NetworkListener, 179
- isSunk
 - structData::Boat, 26
- isValidate
 - guiTable::controllers::PlacementPhaseController, 207
- joinButton
 - guiMain::GameCell, 100
- joinButtonBox
 - guiMain::GameCell, 100
- joinGame
 - guiMain::controller::menuController, 165
 - lo23::battleship::online::network::COMInterface, 61
 - lo23::battleship::online::network::NetworkModule↵
Interface, 183
- JoinGameRequestMessage
 - lo23::battleship::online::network::messages::Join↵
GameRequestMessage, 156
- JoinGameResponseMessage
 - lo23::battleship::online::network::messages::Join↵
GameResponseMessage, 159
- joinGameSpectator
 - data::CDataCom, 36
 - interfacesData::IDataCom, 141
- lastName
 - guiMain::controller::ChangeProfileController, 52
 - guiMain::controller::ProfilController, 219
- lastNameTextField
 - guiMain::controller::SignupController, 246
- lastname
 - structData::Profile, 226
- launchServer
 - lo23::battleship::online::network::NetworkController, 174
- lblj1
 - guiTable::controllers::ObservationPhase, 198
- lblj2
 - guiTable::controllers::ObservationPhase, 198
- listBoats
 - structData::Player, 212
- listCases
 - structData::Boat, 26
- listContacts
 - structData::DataUser, 88
- listGames
 - data::DataController, 85
- listMessages
 - structData::Game, 98
- listShots
 - structData::Player, 212
- listSpectators
 - structData::Game, 98
- listUsers
 - data::DataController, 85
 - structData::ContactGroup, 69
- listener
 - lo23::battleship::online::network::NetworkServer, 191
- lo23, 15
- lo23.battleship, 15

- lo23.battleship.MainApp, 161
- lo23.battleship.online, 15
- lo23.battleship.online.network, 16
- lo23.battleship.online.network.COMInterface, 58
- lo23.battleship.online.network.messages, 16
- lo23.battleship.online.network.messages.Connection↔
 - EstablishedMessage, 64
- lo23.battleship.online.network.messages.Connection↔
 - RequestMessage, 66
- lo23.battleship.online.network.messages.Created↔
 - GameNotificationMessage, 69
- lo23.battleship.online.network.messages.Disconnection↔
 - Message, 88
- lo23.battleship.online.network.messages.GameQuit↔
 - Message, 108
- lo23.battleship.online.network.messages.GameQuit↔
 - SpectatorMessage, 109
- lo23.battleship.online.network.messages.GameWon↔
 - Message, 111
- lo23.battleship.online.network.messages.GameWon↔
 - MessageToSpectator, 113
- lo23.battleship.online.network.messages.GetInfo↔
 - GameForSpectatorMessage, 115
- lo23.battleship.online.network.messages.GetProfile↔
 - RequestMessage, 117
- lo23.battleship.online.network.messages.JoinGame↔
 - RequestMessage, 156
- lo23.battleship.online.network.messages.JoinGame↔
 - ResponseMessage, 158
- lo23.battleship.online.network.messages.Message, 168
- lo23.battleship.online.network.messages.NotifyReady↔
 - Message, 191
- lo23.battleship.online.network.messages.Profile↔
 - RequestAnswerMessage, 227
- lo23.battleship.online.network.messages.SendInfo↔
 - GameForSpectatorMessage, 229
- lo23.battleship.online.network.messages.SendNew↔
 - SpectatorMessage, 231
- lo23.battleship.online.network.messages.SendText↔
 - Message, 233
- lo23.battleship.online.network.messages.ShotNotification↔
 - Message, 237
- lo23.battleship.online.network.messages.ShotNotification↔
 - ResultForSpectatorMessage, 239
- lo23.battleship.online.network.messages.ShotNotification↔
 - ResultMessage, 242
- lo23.battleship.online.network.messages.Update↔
 - GameMessage, 248
- lo23.battleship.online.network.NetworkController, 170
- lo23.battleship.online.network.NetworkListener, 176
- lo23.battleship.online.network.NetworkModuleInterface, 180
- lo23.battleship.online.network.NetworkSender, 186
- lo23.battleship.online.network.NetworkServer, 188
- lo23::battleship::MainApp
 - main, 162
 - start, 162
- lo23::battleship::online::network::COMInterface
 - askDisconnection, 59
 - changeStatusGame, 59
 - clearNetwork, 60
 - coordinates, 60
 - gameQuitSpectator, 60
 - getInfoGameForSpectator, 60
 - getProfile, 61
 - joinGame, 61
 - notifyGameWon, 61
 - notifyJoinGameResponse, 61
 - notifyNewGame, 61
 - notifyReady, 62
 - removeGame, 62
 - searchForPlayers, 62
 - sendChatMessage, 62
 - sendInfoGameForSpectator, 63
 - sendNewSpectator, 63
 - sendShot, 63
- lo23::battleship::online::network::NetworkController
 - addToNetwork, 172
 - addUserToNetwork, 172
 - clearNetwork, 172
 - closeListener, 172
 - dataInterface, 176
 - filterKnownIPAddressesToNotify, 172
 - filterUnknownIPAddresses, 173
 - getAddressForUser, 173
 - getCOMInterface, 173
 - getConnectedUsers, 173
 - getIPTable, 174
 - getInstance, 174
 - getPort, 174
 - instance, 176
 - launchServer, 174
 - NetworkController, 172
 - networkInterface, 176
 - networkServer, 176
 - networkState, 176
 - port, 176
 - removeFromNetwork, 174
 - removeUnreachableHost, 174
 - sendMessage, 174
 - setDataInterface, 176
- lo23::battleship::online::network::NetworkListener
 - closeSocket, 178
 - dataInterface, 179
 - getIsRunning, 178
 - getServerSocketIPAddress, 178
 - isRunning, 179
 - NetworkListener, 178
 - read, 178
 - reader, 179
 - run, 179
 - server, 179
 - serverSocket, 179
 - setDataInterface, 179
 - setIsRunning, 179

- lo23::battleship::online::network::NetworkModule↔
 - Interface
 - askDisconnection, 181
 - changeStatusGame, 182
 - clearNetwork, 182
 - controller, 186
 - coordinates, 182
 - dataInterface, 186
 - gameQuitSpectator, 182
 - getInfoGameForSpectator, 183
 - getProfile, 183
 - joinGame, 183
 - NetworkModuleInterface, 181
 - notifyGameWon, 183
 - notifyJoinGameResponse, 183
 - notifyNewGame, 184
 - notifyReady, 184
 - removeGame, 184
 - searchForPlayers, 184
 - sendChatMessage, 184
 - sendInfoGameForSpectator, 185
 - sendNewSpectator, 185
 - sendShot, 185
 - setDataInterface, 186
- lo23::battleship::online::network::NetworkSender
 - controller, 188
 - host, 188
 - message, 188
 - NetworkSender, 187
 - port, 188
 - run, 188
 - sock, 188
 - writer, 188
- lo23::battleship::online::network::NetworkServer
 - address, 190
 - backlog, 190
 - close, 190
 - dataInterface, 190
 - getIpAddress, 190
 - isOpened, 190
 - listener, 191
 - networkController, 191
 - NetworkServer, 189
 - open, 190
 - port, 191
 - setDataInterface, 190
- lo23::battleship::online::network::messages::Connection↔
 - EstablishedMessage
 - ConnectionEstablishedMessage, 65
 - createdGame, 66
 - getType, 65
 - ipAddressesTable, 66
 - process, 65
 - sender, 66
- lo23::battleship::online::network::messages::Connection↔
 - RequestMessage
 - ConnectionRequestMessage, 67
 - game, 68
 - getType, 67
 - ipAddressesTable, 68
 - process, 67
 - sender, 68
- lo23::battleship::online::network::messages::Created↔
 - GameNotificationMessage
 - CreatedGameNotificationMessage, 70
 - game, 71
 - getType, 70
 - process, 70
- lo23::battleship::online::network::messages::Disconnection↔
 - Message
 - DisconnectionMessage, 89
 - game, 90
 - getType, 89
 - process, 89
 - user, 90
- lo23::battleship::online::network::messages::Game↔
 - QuitMessage
 - game, 109
 - GameQuitMessage, 109
 - getType, 109
 - process, 109
- lo23::battleship::online::network::messages::Game↔
 - QuitSpectatorMessage
 - game, 111
 - GameQuitSpectatorMessage, 110
 - getType, 110
 - process, 110
 - spec, 111
- lo23::battleship::online::network::messages::Game↔
 - WonMessage
 - GameWonMessage, 112
 - getType, 112
 - player, 113
 - process, 112
- lo23::battleship::online::network::messages::Game↔
 - WonMessageToSpectator
 - GameWonMessageToSpectator, 114
 - getType, 114
 - player, 115
 - process, 114
- lo23::battleship::online::network::messages::GetInfo↔
 - GameForSpectatorMessage
 - GetInfoGameForSpectatorMessage, 116
 - getType, 116
 - player, 116
 - process, 116
 - spec, 116
- lo23::battleship::online::network::messages::Get↔
 - ProfileRequestMessage
 - GetProfileRequestMessage, 117
 - getType, 118
 - process, 118
 - userRequester, 118
- lo23::battleship::online::network::messages::Join↔
 - GameRequestMessage
 - game, 157

- getType, 157
 - JoinGameRequestMessage, 156
 - process, 157
 - sender, 157
- lo23::battleship::online::network::messages::Join↔
 - GameResponseMessage
 - game, 160
 - getType, 159
 - isOk, 160
 - JoinGameResponseMessage, 159
 - process, 159
 - sender, 160
- lo23::battleship::online::network::messages::Message
 - getType, 169
 - process, 169
 - type, 169
- lo23::battleship::online::network::messages::Notify↔
 - ReadyMessage
 - destUser, 193
 - getType, 192
 - NotifyReadyMessage, 192
 - process, 192
 - sender, 193
- lo23::battleship::online::network::messages::Profile↔
 - RequestAnswerMessage
 - getType, 228
 - process, 228
 - profile, 229
 - ProfileRequestAnswerMessage, 228
 - user, 229
- lo23::battleship::online::network::messages::SendInfo↔
 - GameForSpectatorMessage
 - game, 231
 - getType, 230
 - process, 230
 - SendInfoGameForSpectatorMessage, 230
 - spec, 231
- lo23::battleship::online::network::messages::Send↔
 - NewSpectatorMessage
 - getType, 232
 - process, 232
 - SendNewSpectatorMessage, 232
 - spec, 233
- lo23::battleship::online::network::messages::Send↔
 - TextMessage
 - getType, 234
 - message, 235
 - process, 234
 - SendTextMessage, 234
- lo23::battleship::online::network::messages::Shot↔
 - NotificationMessage
 - getType, 238
 - process, 238
 - shot, 239
 - ShotNotificationMessage, 238
- lo23::battleship::online::network::messages::Shot↔
 - NotificationResultForSpectatorMessage, 240
- lo23::battleship::online::network::messages::Shot↔
 - NotificationResultMessage
 - boat, 243
 - getType, 243
 - process, 243
 - shot, 243
 - ShotNotificationResultMessage, 242
- lo23::battleship::online::network::messages::Update↔
 - GameMessage
 - gameUpdate, 249
 - getType, 249
 - process, 249
 - UpdateGameMessage, 248
- localGame
 - data::DataController, 85
- localPlayer
 - data::DataController, 85
- localProfile
 - data::DataController, 85
- localUser
 - data::DataController, 85
- logMsg
 - guiTable::controllers::ObservationPhase, 195
 - guiTable::controllers::PlacementPhaseController, 205
 - guiTable::controllers::gameInterface, 101
- logYesNoMsg
 - guiTable::controllers::ObservationPhase, 196
 - guiTable::controllers::gameInterface, 102
- login
 - guiMain::controller::LoginController, 160
 - structData::User, 254
- loginButton
 - guiMain::controller::LoginController, 161
- loginController
 - guiMain::GuiMainController, 125
- lookButton
 - guiMain::GameCell, 100
 - guiMain::PlayerCell, 214
- lookButtonBox
 - guiMain::GameCell, 100
 - guiMain::PlayerCell, 214
- lookGame
 - guiMain::GuiMainController, 122
 - guiMain::controller::menuController, 165
- lookUser
 - guiMain::controller::menuController, 165
- MISSED
 - guiTable::CaseDrawing::Type, 247
- MY_TURN_MSG
 - guiTable::controllers::ObservationPhase, 198
 - guiTable::controllers::gameInterface, 103
- main

- lo23::battleship::MainApp, 162
- mainController
 - guiMain::controller::ChangeProfileController, 52
 - guiMain::controller::CreateGameController, 74
 - guiMain::controller::IpConfigController, 155
 - guiMain::controller::LoginController, 161
 - guiMain::controller::ProfilController, 219
 - guiMain::controller::SignupController, 246
 - guiMain::controller::menuController, 167
- mainStage
 - guiTable::controllers::GuiTableController, 133
- menuController
 - guiMain::GuiMainController, 125
- message
 - lo23::battleship::online::network::NetworkSender, 188
 - lo23::battleship::online::network::messages::SendTextMessage, 235
- messageContainer
 - guiTable::controllers::ObservationPhase, 198
 - guiTable::controllers::PlacementPhaseController, 207
 - guiTable::controllers::gameInterface, 102
- messageLabel
 - guiMain::controller::SignupController, 246
- messageMask
 - guiTable::controllers::GamePhaseController, 107
- messageTextContainer
 - guiTable::controllers::ObservationPhase, 198
 - guiTable::controllers::gameInterface, 102
- modifyAvatar
 - guiMain::controller::ChangeProfileController, 51
- modifyAvatarButton
 - guiMain::controller::ChangeProfileController, 52
- modifyProfileButton
 - guiMain::controller::menuController, 167
- MousePlaceBoat
 - guiTable::controllers::PlacementPhaseController, 205
- myTable
 - guiTable::controllers::GamePhaseController, 107
- myTurn
 - guiTable::controllers::GamePhaseController, 107
- NB_CASES_GRID
 - data::CDataTable, 50
 - guiTable::controllers::BaseController, 20
- NOTIFICATION
 - structData::MessageType, 170
- name
 - guiMain::PlayerCell, 214
 - structData::BoatType, 31
 - structData::ContactGroup, 69
 - structData::Game, 98
 - structData::Profile, 226
- nameTitle
 - guiMain::controller::ChangeProfileController, 52
 - guiMain::controller::ProfilController, 219
- nbCases
 - structData::BoatType, 31
- NetworkController
 - lo23::battleship::online::network::NetworkController, 172
- networkController
 - lo23::battleship::online::network::NetworkServer, 191
- networkInterface
 - lo23::battleship::online::network::NetworkController, 176
- NetworkListener
 - lo23::battleship::online::network::NetworkListener, 178
- NetworkModuleInterface
 - lo23::battleship::online::network::NetworkModuleInterface, 181
- NetworkSender
 - lo23::battleship::online::network::NetworkSender, 187
- NetworkServer
 - lo23::battleship::online::network::NetworkServer, 189
- networkServer
 - lo23::battleship::online::network::NetworkController, 176
- networkState
 - lo23::battleship::online::network::NetworkController, 176
- newGame
 - data::CDataMain, 44
 - guiMain::controller::CreateGameController, 73
 - interfacesData::IDataMain, 148
- newRequestSpectator
 - data::CDataCom, 37
 - interfacesData::IDataCom, 141
- noButton
 - guiTable::controllers::ObservationPhase, 198
 - guiTable::controllers::gameInterface, 103
- noClicked
 - guiTable::controllers::GamePhaseController, 105
 - guiTable::controllers::ObservationPhase, 196
 - guiTable::controllers::ObserverPhaseController, 199
- notifGameChosen
 - data::CDataMain, 45
 - interfacesData::IDataMain, 148
- notifToJoinGame
 - data::CDataCom, 37
 - interfacesData::IDataCom, 141
- notifiedGameWon
 - data::CDataCom, 37
 - interfacesData::IDataCom, 141
- notifyAttendedGameWon
 - data::CDataCom, 37
 - interfacesData::IDataCom, 142
- notifyGameWon
 - lo23::battleship::online::network::COMInterface, 61

- lo23::battleship::online::network::NetworkModule↔
Interface, 183
- notifyJoinGameResponse
 - lo23::battleship::online::network::COMInterface, 61
 - lo23::battleship::online::network::NetworkModule↔
Interface, 183
- notifyNewGame
 - lo23::battleship::online::network::COMInterface, 61
 - lo23::battleship::online::network::NetworkModule↔
Interface, 184
- notifyQuitSpectator
 - data::CDataCom, 37
 - interfacesData::IDataCom, 142
- notifyReady
 - lo23::battleship::online::network::COMInterface, 62
 - lo23::battleship::online::network::NetworkModule↔
Interface, 184
- NotifyReadyMessage
 - lo23::battleship::online::network::messages::↔
NotifyReadyMessage, 192
- notifyToSpecGame
 - data::CDataCom, 38
 - interfacesData::IDataCom, 142
- numberOfGame
 - guiMain::controller::ChangeProfileController, 52
 - guiMain::controller::ProfilController, 219
- numberOfGameLost
 - guiMain::controller::ChangeProfileController, 53
 - guiMain::controller::ProfilController, 219
- numberOfGameWon
 - guiMain::controller::ChangeProfileController, 53
 - guiMain::controller::ProfilController, 219
- OBSERVATION_PHASE_URL
 - guiTable::controllers::GuiTableController, 133
- OBSERVER_PHASE_URL
 - guiTable::controllers::GuiTableController, 133
- OTHER_TURN_MSG
 - guiTable::controllers::ObservationPhase, 198
 - guiTable::controllers::gameInterface, 103
- observationControlleur
 - guiTable::controllers::GuiTableController, 133
- observerPhaseController
 - guiTable::controllers::GuiTableController, 133
- onClickCase
 - guiTable::controllers::GamePhaseController, 105
- onValidate
 - guiTable::controllers::PlacementPhaseController,
205
- open
 - lo23::battleship::online::network::NetworkServer,
190
- openChangeProfileWindow
 - guiMain::GuiMainController, 122
 - guiMain::controller::menuController, 165
- openConfigWindow
 - guiMain::GuiMainController, 122
- openCreateAccountWindow
 - guiMain::controller::LoginController, 161
- openCreateGameWindow
 - guiMain::GuiMainController, 122
 - guiMain::controller::menuController, 166
- openFile
 - guiMain::controller::ChangeProfileController, 51
 - guiMain::controller::SignupController, 245
- openMenuWindow
 - guiMain::GuiMainController, 122
 - guiMain::GuiMainInterface, 127
- openPlacementPhase
 - guiMain::GuiMainController, 122
 - guiMain::GuiMainInterface, 127
- openProfileWindow
 - guiMain::GuiMainController, 122
- openSignupWindow
 - guiMain::GuiMainController, 123
- openWaitingRoomWindow
 - guiMain::GuiMainController, 123
- opponentReady
 - guiTable::GuiTableInterface, 136
 - guiTable::controllers::GuiTableController, 132
- option
 - guiMain::controller::menuController, 166
- optionButton
 - guiMain::controller::menuController, 167
- PLAYER1READY
 - structData::StatusGame, 247
- PLAYER2READY
 - structData::StatusGame, 247
- PLAYING
 - structData::StatusGame, 247
- PORTEAVIONS
 - structData::BoatType, 31
- pane
 - guiMain::IpCell, 154
- password
 - structData::DataUser, 88
- passwordTextField
 - guiMain::controller::SignupController, 246
- placeBoat
 - guiTable::controllers::PlacementPhaseController,
205
- placeShotTo
 - guiTable::controllers::ObservationPhase, 196
 - guiTable::controllers::gameInterface, 102
- placed
 - guiTable::BoatDrawing, 29
- placementPhaseController
 - guiTable::controllers::GuiTableController, 133
- playKeyEvent
 - guiTable::controllers::PlacementPhaseController,
205
- Player
 - structData::Player, 209
- player
 - lo23::battleship::online::network::messages::↔
GameWonMessage, 113

- lo23::battleship::online::network::messages::↔
GameWonMessageToSpectator, 115
 - lo23::battleship::online::network::messages::Get↔
InfoGameForSpectatorMessage, 116
 - lo23::battleship::online::network::messages::↔
ShotNotificationResultForSpectatorMessage,
241
- player1
 - structData::Game, 98
- player1Start
 - structData::Game, 98
- player2
 - structData::Game, 98
- PlayerCell
 - guiMain::PlayerCell, 213
- playersList
 - guiMain::GuiMainController, 125
- playersView
 - guiMain::controller::menuController, 167
- port
 - guiMain::GuiMainController, 125
 - guiMain::controller::lpConfigController, 155
 - lo23::battleship::online::network::NetworkController,
176
 - lo23::battleship::online::network::NetworkSender,
188
 - lo23::battleship::online::network::NetworkServer,
191
 - structData::User, 254
- porteAvionsRectangle
 - guiTable::controllers::ClassicPlacementPhase↔
Controller, 58
- Position
 - structData::Position, 215
- positionCorrect
 - guiTable::controllers::PlacementPhaseController,
206
- positioningTime
 - guiMain::controller::CreateGameController, 74
- process
 - lo23::battleship::online::network::messages::↔
ConnectionEstablishedMessage, 65
 - lo23::battleship::online::network::messages::↔
ConnectionRequestMessage, 67
 - lo23::battleship::online::network::messages::↔
CreatedGameNotificationMessage, 70
 - lo23::battleship::online::network::messages::↔
DisconnectionMessage, 89
 - lo23::battleship::online::network::messages::↔
GameQuitMessage, 109
 - lo23::battleship::online::network::messages::↔
GameQuitSpectatorMessage, 110
 - lo23::battleship::online::network::messages::↔
GameWonMessage, 112
 - lo23::battleship::online::network::messages::↔
GameWonMessageToSpectator, 114
 - lo23::battleship::online::network::messages::Get↔
InfoGameForSpectatorMessage, 116
 - lo23::battleship::online::network::messages::Get↔
ProfileRequestMessage, 118
 - lo23::battleship::online::network::messages::Join↔
GameRequestMessage, 157
 - lo23::battleship::online::network::messages::Join↔
GameResponseMessage, 159
 - lo23::battleship::online::network::messages::↔
Message, 169
 - lo23::battleship::online::network::messages::↔
NotifyReadyMessage, 192
 - lo23::battleship::online::network::messages::↔
ProfileRequestAnswerMessage, 228
 - lo23::battleship::online::network::messages::↔
SendInfoGameForSpectatorMessage, 230
 - lo23::battleship::online::network::messages::↔
SendNewSpectatorMessage, 232
 - lo23::battleship::online::network::messages::↔
SendTextMessage, 234
 - lo23::battleship::online::network::messages::↔
ShotNotificationMessage, 238
 - lo23::battleship::online::network::messages::↔
ShotNotificationResultForSpectatorMessage,
241
 - lo23::battleship::online::network::messages::↔
ShotNotificationResultMessage, 243
 - lo23::battleship::online::network::messages::↔
UpdateGameMessage, 249
- profilController
 - guiMain::GuiMainController, 125
- Profile
 - structData::Profile, 221, 222
- profile
 - lo23::battleship::online::network::messages::↔
ProfileRequestAnswerMessage, 229
 - structData::ChatMessage, 57
 - structData::Player, 212
- ProfileRequestAnswerMessage
 - lo23::battleship::online::network::messages::↔
ProfileRequestAnswerMessage, 228
- profils
 - guiTable::controllers::ChatController, 54
- RANDOM_ROTATION
 - guiTable::controllers::PlacementPhaseController,
207
- README.md, 257
- read
 - lo23::battleship::online::network::NetworkListener,
178
- reader
 - lo23::battleship::online::network::NetworkListener,
179
- ready
 - structData::Player, 212
- receiveAMessage
 - guiTable::controllers::ChatController, 54
- receiveMessage
 - data::CDataCom, 38
 - interfacesData::IDataCom, 142

- receiveReady
 - data::CDataCom, [38](#)
 - interfacesData::IDataCom, [142](#)
- recordVictory
 - data::DataController, [81](#)
- reflectionTime
 - guiMain::controller::CreateGameController, [74](#)
 - guiMain::controller::WaitingRoomController, [255](#)
- refresh
 - guiMain::controller::menuController, [166](#)
- refreshButton
 - guiMain::controller::menuController, [167](#)
- reinit
 - guiTable::BoatDrawing, [28](#)
- reinitBoat
 - guiTable::controllers::PlacementPhaseController, [206](#)
- reloadConversation
 - guiTable::controllers::ChatController, [54](#)
- reloadSavedProfile
 - data::DataController, [81](#)
- removeButton
 - guiMain::lpCell, [154](#)
- removeFromNetwork
 - lo23::battleship::online::network::NetworkController, [174](#)
- removeGame
 - data::CDataCom, [38](#)
 - data::CDataMain, [45](#)
 - guiMain::GuiMainController, [123](#)
 - guiMain::GuiMainInterface, [127](#)
 - guiMain::controller::menuController, [166](#)
 - interfacesData::IDataCom, [143](#)
 - interfacesData::IDataMain, [149](#)
 - lo23::battleship::online::network::COMInterface, [62](#)
 - lo23::battleship::online::network::NetworkModule↔Interface, [184](#)
- removeGameFromList
 - data::CDataCom, [38](#)
 - data::DataController, [81](#)
 - interfacesData::IDataCom, [143](#)
- removeSelectedCase
 - guiTable::controllers::GamePhaseController, [106](#)
- removeUnreachableHost
 - lo23::battleship::online::network::NetworkController, [174](#)
- removeUser
 - data::CDataCom, [39](#)
 - guiMain::GuiMainController, [123](#)
 - guiMain::GuiMainInterface, [127](#)
 - guiMain::controller::menuController, [166](#)
 - interfacesData::IDataCom, [143](#)
- removeUserFromList
 - data::DataController, [81](#)
- returnButton
 - guiMain::controller::CreateGameController, [74](#)
- returnToMenu
 - guiMain::controller::CreateGameController, [73](#)
- rights
 - structData::ContactGroup, [69](#)
- robot
 - guiMain::GameCell, [100](#)
- rootLayout
 - guiMain::GuiMainController, [125](#)
 - guiTable::controllers::GuiTableController, [133](#)
- rotation
 - guiTable::BoatDrawing, [29](#)
- rotationIsValide
 - guiTable::controllers::PlacementPhaseController, [207](#)
- run
 - lo23::battleship::online::network::NetworkListener, [179](#)
 - lo23::battleship::online::network::NetworkSender, [188](#)
- SHOT
 - guiTable::CaseDrawing::Type, [247](#)
- SOUSMARINFR
 - structData::BoatType, [31](#)
- SOUSMARINB
 - structData::BoatType, [31](#)
- SPACE
 - guiTable::controllers::BaseController, [20](#)
- STYLE_MY_TURN
 - guiTable::controllers::ObservationPhase, [198](#)
 - guiTable::controllers::gameInterface, [103](#)
- STYLE_OTHER_TURN
 - guiTable::controllers::ObservationPhase, [198](#)
 - guiTable::controllers::gameInterface, [103](#)
- SUNK_BOAT
 - guiTable::CaseDrawing::Type, [247](#)
- saveButton
 - guiMain::controller::ChangeProfileController, [53](#)
- saveProfile
 - guiMain::controller::ChangeProfileController, [52](#)
 - structData::Profile, [224](#)
- saveeditedProfile
 - structData::Profile, [224](#)
- searchForPlayers
 - lo23::battleship::online::network::COMInterface, [62](#)
 - lo23::battleship::online::network::NetworkModule↔Interface, [184](#)
- selectedCase
 - guiTable::controllers::GamePhaseController, [107](#)
- sendChatMessage
 - lo23::battleship::online::network::COMInterface, [62](#)
 - lo23::battleship::online::network::NetworkModule↔Interface, [184](#)
- sendInfoGameForSpectator
 - lo23::battleship::online::network::COMInterface, [63](#)
 - lo23::battleship::online::network::NetworkModule↔Interface, [185](#)
- SendInfoGameForSpectatorMessage
 - lo23::battleship::online::network::messages::↔SendInfoGameForSpectatorMessage, [230](#)
- sendMessage

- lo23::battleship::online::network::NetworkController, 174
- sendMessage
 - guiTable::controllers::ChatController, 54
- sendNewSpectator
 - lo23::battleship::online::network::COMInterface, 63
 - lo23::battleship::online::network::NetworkModule↔Interface, 185
- SendNewSpectatorMessage
 - lo23::battleship::online::network::messages::↔SendNewSpectatorMessage, 232
- sendShot
 - lo23::battleship::online::network::COMInterface, 63
 - lo23::battleship::online::network::NetworkModule↔Interface, 185
- sendStatistics
 - data::CDataCom, 39
 - guiMain::GuiMainController, 123
 - guiMain::GuiMainInterface, 127
 - interfacesData::IDataCom, 143
- SendTextMessage
 - lo23::battleship::online::network::messages::↔SendTextMessage, 234
- sender
 - lo23::battleship::online::network::messages::↔ConnectionEstablishedMessage, 66
 - lo23::battleship::online::network::messages::↔ConnectionRequestMessage, 68
 - lo23::battleship::online::network::messages::Join↔GameRequestMessage, 157
 - lo23::battleship::online::network::messages::Join↔GameResponseMessage, 160
 - lo23::battleship::online::network::messages::↔NotifyReadyMessage, 193
- serialVersionUID
 - structData::Boat, 26
 - structData::ChatMessage, 57
 - structData::ContactGroup, 69
 - structData::Game, 98
 - structData::Player, 212
 - structData::Position, 217
 - structData::User, 254
- server
 - lo23::battleship::online::network::NetworkListener, 179
- serverSocket
 - lo23::battleship::online::network::NetworkListener, 179
- setActive
 - guiTable::BoatDrawing, 28
- setActiveBoat
 - guiTable::BoatDrawing, 28
- setAttendedGame
 - data::DataController, 81
- setAvatar
 - structData::Profile, 224, 225
- setBirthdate
 - structData::Profile, 225
- setBoatRectangle
 - guiTable::BoatDrawing, 29
- setBoatType
 - guiTable::BoatDrawing, 29
- setContent
 - structData::ChatMessage, 56
- setDataController
 - guiTable::GuiTableInterface, 137
 - guiTable::controllers::ChatController, 54
 - guiTable::controllers::GuiTableController, 132
- setDataInterface
 - lo23::battleship::online::network::NetworkController, 176
 - lo23::battleship::online::network::NetworkListener, 179
 - lo23::battleship::online::network::NetworkModule↔Interface, 186
 - lo23::battleship::online::network::NetworkServer, 190
- setGameJoinResponse
 - data::CDataCom, 39
 - guiMain::GuiMainController, 123
 - guiMain::GuiMainInterface, 127
 - interfacesData::IDataCom, 143, 144
- setGamesAborted
 - structData::Profile, 225
- setGamesLost
 - structData::Profile, 225
- setGamesPlayed
 - structData::Profile, 225
- setGamesWon
 - structData::Profile, 226
- setIPs
 - structData::User, 253
- setIdUser
 - structData::User, 252
- setIdata
 - guiMain::GuiMainController, 124
- setImage
 - guiMain::controller::menuController, 166
- setInterfaceCom
 - data::CDataCom, 40
 - data::CDataMain, 45
 - data::CDataTable, 48
 - data::DataController, 82
- setInterfaceMain
 - data::CDataCom, 40
 - data::CDataTable, 49
 - data::DataController, 82
- setInterfaceTable
 - data::CDataCom, 40
 - data::CDataTable, 49
 - data::DataController, 82
- setIps
 - guiMain::GuiMainController, 124
- setIsRunning
 - lo23::battleship::online::network::NetworkListener, 179

setIsValidate
 guiTable::controllers::PlacementPhaseController,
 206
 setLastName
 structData::Profile, 226
 setLblj1
 guiTable::controllers::ObservationPhase, 196
 setLblj2
 guiTable::controllers::ObservationPhase, 196
 setListBoats
 structData::Player, 211
 setListContacts
 structData::DataUser, 87
 setListGame
 data::DataController, 82
 setListIps
 data::CDataMain, 45
 interfacesData::IDataMain, 149
 setListMessages
 structData::Game, 96
 setListShots
 structData::Player, 211
 setListSpectators
 structData::Game, 97
 setListUser
 data::DataController, 82
 setListcases
 structData::Boat, 25
 setLocalDataUser
 data::DataController, 83
 setLocalGame
 data::CDataCom, 40
 data::CDataMain, 45
 data::DataController, 83
 interfacesData::IDataCom, 144
 interfacesData::IDataMain, 149
 setLocalPlayer
 data::DataController, 83
 setLocalProfile
 data::DataController, 83
 setLocalUser
 data::DataController, 83
 setLogin
 structData::User, 253
 setMainController
 guiMain::controller::ChangeProfileController, 52
 guiMain::controller::CreateGameController, 73
 guiMain::controller::IpConfigController, 155
 guiMain::controller::LoginController, 161
 guiMain::controller::ProfilController, 219
 guiMain::controller::SignupController, 245
 guiMain::controller::menuController, 167
 setMyBoats
 guiTable::controllers::GamePhaseController, 106
 setMyTurn
 guiTable::controllers::GamePhaseController, 106
 setName
 structData::Profile, 226
 setPassword
 structData::DataUser, 88
 setPlaced
 guiTable::BoatDrawing, 29
 setPlacementTime
 guiTable::controllers::PlacementPhaseController,
 206
 setPlayer1
 structData::Game, 97
 setPlayer2
 structData::Game, 97
 setPort
 data::CDataMain, 46
 guiMain::GuiMainController, 124
 interfacesData::IDataMain, 149
 structData::User, 253
 setPosition
 guiTable::BoatDrawing, 29
 setProfil
 guiMain::controller::ProfilController, 219
 setProfile
 structData::ChatMessage, 57
 structData::Player, 211
 setReady
 structData::Player, 212
 setRotation
 guiTable::BoatDrawing, 29
 setRoundTime
 guiTable::controllers::GamePhaseController, 106
 setStage
 guiMain::controller::WaitingRoomController, 255
 setStatus
 structData::Game, 97
 setSunk
 structData::Boat, 25
 setTableController
 guiTable::controllers::ObservationPhase, 196
 setTime
 structData::ChatMessage, 57
 structData::Shot, 236
 setTouched
 structData::Position, 216
 setTurn
 guiTable::controllers::ObservationPhase, 196
 guiTable::controllers::ObserverPhaseController,
 199
 setType
 structData::Boat, 25
 setUsername
 structData::User, 253
 setX
 structData::Position, 216
 setY
 structData::Position, 217
 Shot
 structData::Shot, 236
 shot

- lo23::battleship::online::network::messages::↔
ShotNotificationMessage, 239
 - lo23::battleship::online::network::messages::↔
ShotNotificationResultForSpectatorMessage,
241
 - lo23::battleship::online::network::messages::↔
ShotNotificationResultMessage, 243
- ShotNotificationMessage
 - lo23::battleship::online::network::messages::↔
ShotNotificationMessage, 238
- ShotNotificationResultForSpectatorMessage
 - lo23::battleship::online::network::messages::↔
ShotNotificationResultForSpectatorMessage,
240
- ShotNotificationResultMessage
 - lo23::battleship::online::network::messages::↔
ShotNotificationResultMessage, 242
- showDefeat
 - guiTable::controllers::GamePhaseController, 106
 - guiTable::controllers::gameInterface, 102
- showVictory
 - guiTable::controllers::GamePhaseController, 106
 - guiTable::controllers::ObservationPhase, 197
 - guiTable::controllers::gameInterface, 102
- signUpController
 - guiMain::GuiMainController, 125
- sock
 - lo23::battleship::online::network::NetworkSender,
188
- sousMarinRectangle
 - guiTable::controllers::ClassicPlacementPhase↔
Controller, 58
- sousMarinRectangle1
 - guiTable::controllers::BelgianPlacementPhase↔
Controller, 22
- sousMarinRectangle2
 - guiTable::controllers::BelgianPlacementPhase↔
Controller, 22
- sousMarinRectangle3
 - guiTable::controllers::BelgianPlacementPhase↔
Controller, 22
- sousMarinRectangle4
 - guiTable::controllers::BelgianPlacementPhase↔
Controller, 22
- spec
 - lo23::battleship::online::network::messages::↔
GameQuitSpectatorMessage, 111
 - lo23::battleship::online::network::messages::Get↔
InfoGameForSpectatorMessage, 116
 - lo23::battleship::online::network::messages::↔
SendInfoGameForSpectatorMessage, 231
 - lo23::battleship::online::network::messages::↔
SendNewSpectatorMessage, 233
- spectator
 - structData::Game, 98
- spectatorAutorisation
 - guiMain::controller::WaitingRoomController, 255
- spectatorChat
 - structData::Game, 98
- spectatorsAutorise
 - guiMain::controller::CreateGameController, 74
- src/main/java/data/CDataCom.java, 257
- src/main/java/data/CDataMain.java, 257
- src/main/java/data/CDataTable.java, 257
- src/main/java/data/DataController.java, 258
- src/main/java/guiMain/GameCell.java, 260
- src/main/java/guiMain/GuiMainController.java, 260
- src/main/java/guiMain/GuiMainInterface.java, 260
- src/main/java/guiMain/IpCell.java, 261
- src/main/java/guiMain/PlayerCell.java, 261
- src/main/java/guiMain/controller/ChangeProfile↔
Controller.java, 258
- src/main/java/guiMain/controller/CreateGameController.↔
java, 258
- src/main/java/guiMain/controller/IpConfigController.java,
258
- src/main/java/guiMain/controller/LoginController.java,
259
- src/main/java/guiMain/controller/ProfilController.java,
259
- src/main/java/guiMain/controller/SignupController.java,
259
- src/main/java/guiMain/controller/WaitingRoomController.↔
java, 260
- src/main/java/guiMain/controller/menuController.java,
259
- src/main/java/guiTable/BoatDrawing.java, 261
- src/main/java/guiTable/CaseDrawing.java, 261
- src/main/java/guiTable/GuiTableInterface.java, 264
- src/main/java/guiTable/controllers/BaseController.java,
262
- src/main/java/guiTable/controllers/BelgianPlacement↔
PhaseController.java, 262
- src/main/java/guiTable/controllers/ChatController.java,
262
- src/main/java/guiTable/controllers/ClassicPlacement↔
PhaseController.java, 262
- src/main/java/guiTable/controllers/GamePhase↔
Controller.java, 263
- src/main/java/guiTable/controllers/GuiTableController.↔
java, 263
- src/main/java/guiTable/controllers/ObservationPhase.↔
java, 263
- src/main/java/guiTable/controllers/ObserverPhase↔
Controller.java, 264
- src/main/java/guiTable/controllers/PlacementPhase↔
Controller.java, 264
- src/main/java/guiTable/controllers/ProfileController.java,
264
- src/main/java/guiTable/controllers/gameInterface.java,
263
- src/main/java/interfacesData/IDataCom.java, 265
- src/main/java/interfacesData/IDataMain.java, 265
- src/main/java/interfacesData/IDataTable.java, 265
- src/main/java/lo23/battleship/MainApp.java, 265
- src/main/java/lo23/battleship/online/network/COM↔

Interface.java, 266
 src/main/java/lo23/battleship/online/network/NetworkController.java, 274
 src/main/java/lo23/battleship/online/network/NetworkListener.java, 274
 src/main/java/lo23/battleship/online/network/NetworkModuleInterface.java, 274
 src/main/java/lo23/battleship/online/network/NetworkSender.java, 274
 src/main/java/lo23/battleship/online/network/NetworkServer.java, 275
 src/main/java/lo23/battleship/online/network/messages/ConnectionEstablishedMessage.java, 266
 src/main/java/lo23/battleship/online/network/messages/ConnectionRequestMessage.java, 266
 src/main/java/lo23/battleship/online/network/messages/CreatedGameNotificationMessage.java, 267
 src/main/java/lo23/battleship/online/network/messages/DisconnectionMessage.java, 267
 src/main/java/lo23/battleship/online/network/messages/GameQuitMessage.java, 267
 src/main/java/lo23/battleship/online/network/messages/GameQuitSpectatorMessage.java, 268
 src/main/java/lo23/battleship/online/network/messages/GameWonMessage.java, 268
 src/main/java/lo23/battleship/online/network/messages/GameWonMessageToSpectator.java, 268
 src/main/java/lo23/battleship/online/network/messages/GetInfoGameForSpectatorMessage.java, 269
 src/main/java/lo23/battleship/online/network/messages/GetProfileRequestMessage.java, 269
 src/main/java/lo23/battleship/online/network/messages/JoinGameRequestMessage.java, 269
 src/main/java/lo23/battleship/online/network/messages/JoinGameResponseMessage.java, 270
 src/main/java/lo23/battleship/online/network/messages/Message.java, 270
 src/main/java/lo23/battleship/online/network/messages/NotifyReadyMessage.java, 270
 src/main/java/lo23/battleship/online/network/messages/ProfileRequestAnswerMessage.java, 271
 src/main/java/lo23/battleship/online/network/messages/SendInfoGameForSpectatorMessage.java, 271
 src/main/java/lo23/battleship/online/network/messages/SendNewSpectatorMessage.java, 272
 src/main/java/lo23/battleship/online/network/messages/SendTextMessage.java, 272
 src/main/java/lo23/battleship/online/network/messages/ShotNotificationMessage.java, 272
 src/main/java/lo23/battleship/online/network/messages/ShotNotificationResultForSpectatorMessage.java, 273
 src/main/java/lo23/battleship/online/network/messages/ShotNotificationResultMessage.java, 273
 src/main/java/lo23/battleship/online/network/messages/UpdateGameMessage.java, 273
 src/main/java/lo23/battleship/online/network/messages/package-info.java, 271
 src/main/java/lo23/battleship/online/network/package-info.java, 271
 src/main/java/structData/Boat.java, 275
 src/main/java/structData/BoatType.java, 275
 src/main/java/structData/ChatMessage.java, 276
 src/main/java/structData/ContactGroup.java, 276
 src/main/java/structData/DataUser.java, 276
 src/main/java/structData/Game.java, 276
 src/main/java/structData/MessageType.java, 277
 src/main/java/structData/Player.java, 277
 src/main/java/structData/Position.java, 277
 src/main/java/structData/Profile.java, 277
 src/main/java/structData/Shot.java, 278
 src/main/java/structData/StatusGame.java, 278
 src/main/java/structData/User.java, 278
 stage
 guiMain::GuiMainController, 125
 guiMain::controller::WaitingRoomController, 255
 start
 lo23::battleship::MainApp, 162
 startIHM
 guiMain::GuiMainController, 124
 status
 structData::Game, 98
 structData, 18
 structData.Boat, 22
 structData.BoatType, 30
 structData.ChatMessage, 54
 structData.ContactGroup, 68
 structData.DataUser, 85
 structData.Game, 90
 structData.MessageType, 169
 structData.Player, 208
 structData.Position, 214
 structData.Profile, 220
 structData.Shot, 235
 structData.StatusGame, 246
 structData.User, 250
 structData::Boat
 addPosition, 24
 Boat, 23, 24
 cloneBoat, 24
 getListCases, 24
 getSunk, 24
 getType, 25
 isSunk, 26
 listCases, 26
 serialVersionUID, 26
 setListcases, 25
 setSunk, 25
 setType, 25
 type, 26
 updateShot, 26
 verifyBoatStatus, 26
 structData::BoatType
 BoatType, 30

- CROISEURFR, 31
- CROISEURB, 31
- CUIRASSE, 31
- getName, 30
- getNbCases, 30
- name, 31
- nbCases, 31
- PORTEAVIONS, 31
- SOUSMARINFR, 31
- SOUSMARINB, 31
- TORPILLEUR, 31
- structData::ChatMessage
 - ChatMessage, 55
 - cloneChatMessage, 56
 - content, 57
 - getContent, 56
 - getProfile, 56
 - getTime, 56
 - profile, 57
 - serialVersionUID, 57
 - setContent, 56
 - setProfile, 57
 - setTime, 57
 - time, 57
- structData::ContactGroup
 - ContactGroup, 69
 - listUsers, 69
 - name, 69
 - rights, 69
 - serialVersionUID, 69
- structData::DataUser
 - cloneDataUser, 87
 - DataUser, 86, 87
 - getListContacts, 87
 - getPassword, 87
 - listContacts, 88
 - password, 88
 - setListContacts, 87
 - setPassword, 88
- structData::Game
 - addMessage, 93
 - addSpectators, 94
 - classicType, 97
 - clone, 94
 - doesProfileBelongToGame, 94
 - Game, 92, 93
 - getClassicType, 94
 - getHumanOpponent, 94
 - getIdGame, 94
 - getListMessages, 95
 - getListSpectators, 95
 - getName, 95
 - getPlayer1, 95
 - getPlayer1Start, 95
 - getPlayer2, 95
 - getSpectator, 96
 - getSpectatorChat, 96
 - getStatus, 96
 - getTimePerShot, 96
 - getTimeToPlaceBoats, 96
 - humanOpponent, 97
 - idGame, 97
 - listMessages, 98
 - listSpectators, 98
 - name, 98
 - player1, 98
 - player1Start, 98
 - player2, 98
 - serialVersionUID, 98
 - setListMessages, 96
 - setListSpectators, 97
 - setPlayer1, 97
 - setPlayer2, 97
 - setStatus, 97
 - spectator, 98
 - spectatorChat, 98
 - status, 98
 - timePerShot, 98
 - timeToPlaceBoats, 98
- structData::MessageType
 - ERROR, 170
 - NOTIFICATION, 170
- structData::Player
 - addBoat, 209
 - addShot, 210
 - clonePlayer, 210
 - compareProfileToPlayer, 210
 - getListBoats, 210
 - getListShots, 210
 - getProfile, 211
 - isReady, 211
 - listBoats, 212
 - listShots, 212
 - Player, 209
 - profile, 212
 - ready, 212
 - serialVersionUID, 212
 - setListBoats, 211
 - setListShots, 211
 - setProfile, 211
 - setReady, 212
- structData::Position
 - clone, 215
 - getTouched, 216
 - getX, 216
 - getY, 216
 - Position, 215
 - serialVersionUID, 217
 - setTouched, 216
 - setX, 216
 - setY, 217
 - touched, 217
 - x, 217
 - y, 217
- structData::Profile
 - avatar, 226

- birthdate, [226](#)
- cleanUp, [222](#)
- clone, [222](#)
- gamesAborted, [226](#)
- gamesLost, [226](#)
- gamesPlayed, [226](#)
- gamesWon, [226](#)
- getAvatar, [223](#)
- getBirthdate, [223](#)
- getGamesAborted, [223](#)
- getGamesLost, [223](#)
- getGamesPlayed, [223](#)
- getGamesWon, [223](#)
- getImage, [224](#)
- getLastname, [224](#)
- getName, [224](#)
- lastname, [226](#)
- name, [226](#)
- Profile, [221](#), [222](#)
- saveProfile, [224](#)
- saveeditedProfile, [224](#)
- setAvatar, [224](#), [225](#)
- setBirthdate, [225](#)
- setGamesAborted, [225](#)
- setGamesLost, [225](#)
- setGamesPlayed, [225](#)
- setGamesWon, [226](#)
- setLastname, [226](#)
- setName, [226](#)
- structData::Shot
 - getTime, [236](#)
 - setTime, [236](#)
 - Shot, [236](#)
 - time, [237](#)
- structData::StatusGame
 - BOATPHASE, [247](#)
 - FINISHED, [247](#)
 - PLAYER1READY, [247](#)
 - PLAYER2READY, [247](#)
 - PLAYING, [247](#)
 - WAITINGBOT, [247](#)
 - WAITINGPLAYER, [247](#)
- structData::User
 - cloneUser, [251](#)
 - DEFAULT_PORT, [253](#)
 - getIPs, [252](#)
 - getIdUser, [252](#)
 - getLogin, [252](#)
 - getPort, [252](#)
 - getUsername, [252](#)
 - IPs, [253](#)
 - idUser, [253](#)
 - login, [254](#)
 - port, [254](#)
 - serialVersionUID, [254](#)
 - setIPs, [253](#)
 - setIdUser, [252](#)
 - setLogin, [253](#)
 - setPort, [253](#)
 - setUsername, [253](#)
 - User, [251](#)
 - username, [254](#)
- sunckBoat
 - guiTable::controllers::GamePhaseController, [106](#)
- sunckABoat
 - guiTable::controllers::ObservationPhase, [197](#)
 - guiTable::controllers::gameInterface, [102](#)
- sunckMyBoat
 - guiTable::controllers::GamePhaseController, [107](#)
- sunckPlayerBoat
 - guiTable::GuiTableInterface, [137](#)
 - guiTable::controllers::GuiTableController, [132](#)
 - guiTable::controllers::ObservationPhase, [197](#)
 - guiTable::controllers::ObserverPhaseController, [200](#)
- TITLE
 - guiTable::controllers::GuiTableController, [133](#)
- TORPILLEUR
 - structData::BoatType, [31](#)
- TOUCHED
 - guiTable::CaseDrawing::Type, [247](#)
- table
 - guiTable::controllers::GamePhaseController, [107](#)
 - guiTable::controllers::PlacementPhaseController, [207](#)
- table1
 - guiTable::controllers::ObservationPhase, [198](#)
 - guiTable::controllers::ObserverPhaseController, [200](#)
- table2
 - guiTable::controllers::ObservationPhase, [198](#)
 - guiTable::controllers::ObserverPhaseController, [200](#)
- tableController
 - guiTable::controllers::ObservationPhase, [198](#)
 - guiTable::controllers::gameInterface, [103](#)
- tablePlayer
 - guiTable::controllers::ObservationPhase, [198](#)
 - guiTable::controllers::ObserverPhaseController, [200](#)
- testShot
 - data::DataController, [84](#)
- textMessage
 - data::CDataTable, [49](#)
 - interfacesData::IDataTable, [152](#)
- time
 - guiTable::controllers::GamePhaseController, [107](#)
 - guiTable::controllers::PlacementPhaseController, [207](#)
 - structData::ChatMessage, [57](#)
 - structData::Shot, [237](#)
- timeDisabled
 - guiMain::controller::CreateGameController, [74](#)
- timelsOver
 - guiTable::controllers::GamePhaseController, [107](#)

- guiTable::controllers::PlacementPhaseController, 206
- timePerShot
 - guiTable::controllers::GamePhaseController, 107
 - structData::Game, 98
- timePlacement
 - guiTable::controllers::PlacementPhaseController, 207
- timeToPlaceBoats
 - structData::Game, 98
- timeline
 - guiTable::controllers::GamePhaseController, 107
 - guiTable::controllers::PlacementPhaseController, 207
- timerDisable
 - guiMain::controller::CreateGameController, 73
- timerLabel
 - guiTable::controllers::GamePhaseController, 107
 - guiTable::controllers::ObservationPhase, 198
 - guiTable::controllers::PlacementPhaseController, 207
- timerOver
 - data::CDataTable, 49
 - interfacesData::IDataTable, 152
- title
 - guiMain::GameCell, 100
- torpilleurRectangle
 - guiTable::controllers::ClassicPlacementPhaseController, 58
- torpilleurRectangle1
 - guiTable::controllers::BelgianPlacementPhaseController, 22
- torpilleurRectangle2
 - guiTable::controllers::BelgianPlacementPhaseController, 22
- torpilleurRectangle3
 - guiTable::controllers::BelgianPlacementPhaseController, 22
- touched
 - structData::Position, 217
- transmitNewStatus
 - guiMain::GuiMainController, 124
 - guiMain::GuiMainInterface, 128
- turn
 - guiTable::controllers::ObservationPhase, 198
- type
 - guiMain::GameCell, 100
 - lo23::battleship::online::network::messages::Message, 169
 - structData::Boat, 26
- updateAttendedGame
 - data::CDataCom, 40
 - interfacesData::IDataCom, 144
- updateGameData
 - data::DataController, 84
- updateGameDataPlaying
 - data::DataController, 84
- UpdateGameMessage
 - lo23::battleship::online::network::messages::UpdateGameMessage, 248
- updateGameStatus
 - data::DataController, 84
 - guiMain::controller::menuController, 167
- updateItem
 - guiMain::GameCell, 99
 - guiMain::IpCell, 153
 - guiMain::PlayerCell, 213
- updateShot
 - structData::Boat, 26
- updateSpectatorGame
 - guiTable::GuiTableInterface, 137
 - guiTable::controllers::GuiTableController, 132
- User
 - structData::User, 251
- user
 - lo23::battleship::online::network::messages::DisconnectionMessage, 90
 - lo23::battleship::online::network::messages::ProfileRequestAnswerMessage, 229
- userAvatar
 - guiMain::controller::ChangeProfileController, 53
 - guiMain::controller::ProfilController, 220
- userLogin
 - guiMain::controller::LoginController, 161
- userName
 - guiMain::controller::ChangeProfileController, 53
 - guiMain::controller::ProfilController, 220
- userPassword
 - guiMain::controller::ChangeProfileController, 53
 - guiMain::controller::LoginController, 161
- userRequester
 - lo23::battleship::online::network::messages::GetProfileRequestMessage, 118
- username
 - structData::User, 254
- usernameTextField
 - guiMain::controller::SignupController, 246
- VICTORY_MSG
 - guiTable::controllers::gameInterface, 103
- VICTORY_P1_MSG
 - guiTable::controllers::ObservationPhase, 198
- VICTORY_P2_MSG
 - guiTable::controllers::ObservationPhase, 198
- validate
 - guiMain::controller::IpConfigController, 155
- validateBoats
 - guiTable::controllers::GuiTableController, 132
- validateButton
 - guiMain::controller::IpConfigController, 155
- validateShot
 - guiTable::controllers::GamePhaseController, 107
 - guiTable::controllers::GuiTableController, 132
- valider
 - guiTable::controllers::GamePhaseController, 107
 - guiTable::controllers::PlacementPhaseController, 207

- verifyBoatStatus
 - structData::Boat, [26](#)
- WAITINGBOT
 - structData::StatusGame, [247](#)
- WAITINGPLAYER
 - structData::StatusGame, [247](#)
- waitExit
 - guiTable::controllers::ObservationPhase, [198](#)
 - guiTable::controllers::gameInterface, [103](#)
- waitMsg
 - guiTable::controllers::ObservationPhase, [198](#)
 - guiTable::controllers::gameInterface, [103](#)
- waitingRoomController
 - guiMain::GuiMainController, [125](#)
- warningLabel
 - guiMain::controller::menuController, [167](#)
- writer
 - lo23::battleship::online::network::NetworkSender,
[188](#)
- x
 - structData::Position, [217](#)
- y
 - structData::Position, [217](#)
- yesButton
 - guiTable::controllers::ObservationPhase, [198](#)
 - guiTable::controllers::gameInterface, [103](#)
- yesClicked
 - guiTable::controllers::ObservationPhase, [197](#)
 - guiTable::controllers::gameInterface, [102](#)